DISKRESIONêRE SEKERHEID IN OBJEK GEORIëNTEERDE OMGEWINGS

deur

PHILIPINA WILHELMINA JANSEN VAN RENSBURG

VERHANDELING

Voorgelê ter vervulling van die vereistes vir
die graad

MAGISTER IN DIE NATUURWETENSKAPPE

in

Rekenaarwetenskap

in die

FAKULTEIT NATUURWETENSKAPPE

aan die

RANDSE AFRIKAANSE UNIVERSITEIT

Studieleier : Dr. MS Olivier

NOVEMBER 1994

INHOUDSOPGAWE

HOOFSTUK 1 1

DOELSTELLING 1

1.1. INLEIDING 1

1.2. PROBLEEMSTELLING 1

1.3. DEFINISIES 3

1.4. AFSLUITING 4

HOOFSTUK 2 8

ALGEMENE SEKERHEID 8

2.1. INLEIDING 8

2.2. REKENAARSEKERHEID : OORSPRONG EN OORSAKE 9

2.3. VERKRYGING VAN REKENAARSEKERHEID. 14

2.3.1. DIE SEKERHEIDSPLAN. 16

2.3.2. DIE SEKERHEIDSBELEID. 17

2.3.2.1. SEKERHEIDSMODELLE

2.4. EVALUERING VAN SEKERHEIDSTELSELS. 31

2.5. NETWERK TCSEC (TNI) 39

2.5.1. DIE NETWERK BETROUBARE REKENAARBASIS (NBRB) 41

2.6. ITSEC 44

GEVOLGTREKKING 46

HOOFSTUK 3 48

REKENAARSEKERHEID - GRONDBEGINSELS EN MEGANISMES 48

3.1. INLEIDING 48

3.2.SEKERHEIDMEGANISMES 48

3.3. TERMINOLOGIE 49

3.4. BESKERMINGSMEGANISMES - TOEGANGSBEHEER TOT ENTITEITE 51

3.4.1. GIDSLYSTE 51

3.4.2. TOEGANGSBEHEERLYS 53

3.4.3. TOEGANGSBEHEERMATRIKSE 54

3.4.3.1. EKSKLUSIEF-UITSLUITENDE TOEGANGSBEHEERMATRIKS 55

3.4.4. VERMOëGEBASEERDE MEGANISMES 56

3.4.5. PROSEDURE-GEORIëNTEERDE TOEGANGSBEHEER 61
3.4.6. SLOT- EN SLEUTELMEGANISMES 61
3.4.7. KRIPTOGRAFIE 61
3.5. DATABASISSEKERHEIDSMEGANISMES 63
3.5.1. GESENTRALISEERDE BEHEER TEENOOR GEDESENTRALISEERDE BEHEER. 63
3.5.2. EIENAARSKAP TEENOOR ADMINISTRASIE 63
3.5.3. BELEID VIR TOEGANGSBEHEERSPESIFIKASIE 64
3.5.4. BELEID OM INLIGTINGSVLOEI TE BEHEER. 65
3.6. . BESKERMINGSMEGANISMES - GEHEUEBESKERMING EN ADRESSERING 66
3.6.1. Heining/Grens-tegniek. 66
3.6.2. Heralokering 67
3.6.3. Basis/Grens-registers. 67
3.6.4. Geëtiketeerde Argitektuur 67
3.6.5. Segmentering 68
3.6.7.Paginering 69
3.7. SERTIFISERING 69
3.7.1. WAGWOORDE 69
3.8. GEVOLGTREKKING 70

HOOFSTUK 4 72
DIE OBJEKGEORIëNTEERDE PARADIGMA 72
4.1. INLEIDING 72
4.2. GRONDBEGINSELS 73
4.1.1. OBJEKTE 74
4.2.2. KLASSE 77
4.1.3. METODES 81
4.1.4. BOODSKAPPE 83
4.1.5. VOORKOMS 86
4.1.6. BINDING 86
4.1.8. ABSTRAKTE KLASSE 88
4.1.9. OORERWING 88
4.1.10 ENKAPSULERING 90
4.1.11 POLIMORFISMES 91
4.1.12 ABSTRAKSIES 91
GEVOLGTREKKING 93

HOOFSTUK 5 94

RIGLYNE VIR DIE BOU VAN 'N OBJEKGEORIëNTEERDE SEKERHEIDSMODEL

5.1.INLEIDING 94

5.2. DIE UITEENSETTING VAN 'N MODEL 97

5.3. BELEIDSRIGTINGE 104

5.3.1. ALGEMENE DATABASISSTELSELBELEIDSRIGTINGE 104

5.3.1.1. OOP VS GESLOTE STELSELS 105

5.3.1.2. EIENAARSKAP VS ADMINISTRASIE 105

5.3.1.3. DISKRESIONêRE VS MULTIVLAKSEKERHEID 106

5.3.1.4. GREIN. 107

5.3.1.5. INTEGRITEITSEKERHEIDSBELEID 108

5.3.2. BELEID VIR OBJEKGEORIENTEERDE DATABASIS STELSELS 108

5.3.2.1. OORERWING 110

5.3.2.2. SIGBAARHEID VAN ONDER 110

5.3.2.3. SIGBAARHEID VAN BO 111

5.3.2.4. NEGATIEWE MAGTIGING 113

IMPLISIETE SPESIFIEKHEID 115

PREDIKATE 115

5.3.2.5 DATAKLASSIFIKASIEBELEID 116

5.3.3. BELEID MET BETREKKING OP ADMINISTRATIEWE REGTE 117

5.4. UITLEG VAN DIE MODEL 118

5.4.1. PASSIEWE ELEMENTE VAN DIE MODEL. 119

5.4.2. ELEMENT VAN BESKERMING. 119

5.4.3. AKTIEWE ELEMENTE VAN DIE MODEL 122

5.4.4. WISSELWERKING TUSSEN AKTIEWE EN PASSIEWE ELEMENTE 124

5.4.4.1. DIE KOMBINASIE AS MAGTIGINGSBASIS 124

5.4.4.2. DISKRESIONêRE SEKERHEIDSTOEGANGSBEHEER MEGANISME. 128

5.4.4.2. VERPLIGTE SEKERHEIDSTOEGANGSBEHEER MEGANISME 131

5.4.5. DIE VERMOë AS MAGTIGINGSMEGANISME 134

GEVOLGTREKKING 135

HOOFSTUK 6 **136**

DIE DISKRESIONêRE SEKERHEIDSMODEL (DISMOD)

6.1. DOELWIT VAN DISMOD 136
6.2. KOMPONENTE VAN DISMOD 138
6.2.1. DIE ENTITEIT 138
6.2.2. SUBJEKTE 139
6.2.3. DIE VERMOë 141
6.2.4. DIE STELSELSEKERHEIDSBEAMPTE(SSB) 145
AFSLUITING 146

HOOFSTUK 7 **150**

DIE WERKING VAN DISMOD **150**

7.1. INLEIDING 150
7.2. BEHEER OOR DIE SKEPPING VAN ENTITEITE. 150
7.3. BEHEER OOR DIE UITDELING VAN VERMOëNS 158
DIE UITDEEL VAN VERMOëNS 159
DIE OUDITAREA 160
Die UITDEEL VAN DIE UITDEELREG 161
7.4. KONTROLE OOR DIE WEGNEEM VAN VERMOëNS 163
7.5. DIE GEBRUIK VAN ENTITEITE SONDER VERMOëNS 164
7.6. BEHEER MOONTLIKHEDE VAN DIE EIENAAR VAN 'N ENTITEIT. 165
AFSLUITING 166

HOOFSTUK 8 **167**

DIE WERKING VAN DISMOD (VERVOLG)

8.1. INLEIDING 167
8.2. Die STELSEL SEKERHEIDSBEAMPTE 167
8.3. DINAMIESE BINDING 168
8.4. DIE GEBRUIK VAN SKADUKOPIEë 170
8.5. DIE GEBRUIK VAN ROLLE. 171
8.6. GEVOLGTREKKING 171

AFSLUITING. **172**

SUMMARY

Discretionary security is one of the areas in computer security that still needs a lot of attention, therefore this study investigated the use of discretionary security with specific reference to object oriented databases.

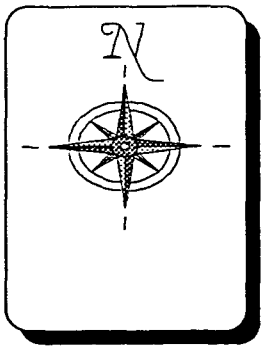
As starting point the basics of computer security, especially security models and mechanisms, as well as the basics of object orientation will be discussed. The study then continues to give guidelines for the building of an object oriented discretionary security model. These guidelines are based on the application of different mechanisms in currently used discretionary object oriented security models, for example. DISCO, DAMOKLES, The MODEL FOR NEXT GENERATION DATABASES, SODA, etc.

A New discretionary object oriented security model - DISMOD - is then introduced . DISMOD attempts to improve on existing models by giving a finer grain of security, as well as a more flexible security system. The model is based on the use of the capability security mechanism, but capabilities in this case are objects (in the object oriented view), which means that a greater functionality can be built into the capabilities. The "Need-to-know" rule can be applied much neater by the owner of entities. The owner of the entities now has the ability to specify capabilities as keys to his entities in such a way that he can give keys to the other users or subjects to suit their needs. The model can be implemented in an object oriented database, and to be most effective it should be part of a trusted computing base.

HOOFSTUK 1

DOELSTELLING

1.1. INLEIDING



Rekenaarsekerheid speel 'n baie belangrike rol in byna elke organisasie vandag as gevolg van die hoeveelheid wandade wat gepleeg word met behulp van die rekenaar. Die feit dat hierdie tipe dade verontskuldig word as gevolg van 'n tekort aan genoegsame bewyse om sodanige dade uit te wys, veroorsaak dat sekerheidstelsels al hoe meer doeltreffend en effektief moet wees om enige wandade te verhoed. Die ontwikkeling van nuwe metodologieë en tegnologieë veroorsaak ook dat nuwe ontwikkeling van rekenaarstelsels nie deurentyd deur die bestaande sekerheidstelsels ondersteun kan word nie. Verskeie tekorkominge ontstaan in die bestaande sekerheidstelsels as gevolg van hierdie nuwe tegnologieë en metodologieë wat oorbrug moet word om volledige, betroubare en effektiewe sekerheidstelsels daar te stel.

Die doel van die verhandeling is om 'n volledige, betroubare en effektiewe sekerheidstelsel daar te stel wat die huidige tegnologie sal ondersteun en ook voorsiening sal maak vir die voorkoming van enige oortreding. Die doel en uiteensetting van die verhandeling sal vervolgens verduidelik word.

1.2. PROBLEEMSTELLING

Daar bestaan 'n groot verskeidenheid sekerheidsmodelle wat ontstaan het as gevolg van die tekort in sekerheidsmodelle wat die huidige rekenaartegnologie ondersteun. Elk van hierdie modelle het egter sy voor- en nadele waarop daar uitgebou of verbeter kan word. Sekerheidsmodelle word meestal gebaseer op die Bell-Lapadulamodel[Pfl89] en gebruik ook meestal verpligte sekerheidsmeganismes as basis van die sekerheidsmodel. Diskresionêre sekerheidsmeganismes word meestal as hulpmiddel ingespan by die gewone sekerheidsmodel. Die gebruik van diskresionêre sekerheidsmeganismes as basis vir 'n sekerheidsmodel bied egter legio navorsingsmoontlikhede. Die kombinering van diskresionêre sekerheid en die

objekgeoriënteerde paradigma blyk om 'n goeie kombinasie te wees in die bou van 'n sekerheidsmodel, daarom word daar in hierdie verhandeling veral aandag geskenk aan 'n sekerheidsmodel van hierdie tipe. DISMOD, 'n diskresionêre sekerheidsmodel, is 'n uitvloeisel van hierdie idee en moet voldoen aan die volgende vereistes:

A Die model moet *buigbaar*, *aanpasbaar* en *doeltreffend* wees,

B Die model moet gebruik kan word in enige *objekgeoriënteerde omgewing*,

C *Vermoëns* - 'n tipe sleutel (verduidelikings volg) moet as toegangsmeganisme gebruik word. Dié vermoëns moet *programmeerbaar* wees sodat alle inligting wat nodig is in die sleutel deur 'n vermoë omvat kan word.

D 'n Fyner mate van beheer moet ingestel word, sodanig so dat enige tipe entiteit tot in die fynste besonderhede beskerm sal word, d.w.s. die semantiek van die data moet ook byvoorbeeld tot in die fynste besonderhede ondersteun kan word,

E 'n *Rol-tipe-beskerming* moet ondersteun kan word. Rol-tipe-beskerming is waar beskerming van entiteite gedoen word op 'n rolbasis. Toegangsbeheer word byvoorbeeld gedoen op rolle eerder as subjekte. 'n Voorbeeld van 'n gebruikersrol is die naam wat toegeken word aan sekere funksies wat uitgevoer word in 'n maatskappy, soos byvoorbeeld 'n sekretaresse, ens. Individue soos die databasisadministrateur, die projekleier, moet ook in staat wees om hul eie rolle te spesifiseer vir die omgewing waarin hulle funksioneer. Die rolle wat deur só 'n individu gedefinieer is, moet deur daardie individu gebruik kan word in die uitoefening van sekerheid vir entiteite in sy omgewing. Dit kan gebruik word in die spesifisering van toegangsbeheer deur subjekte of gebruikers in die spesifisering van die toegang te vervang met die spesifieke rol, d.w.s. die toegang word nou toegeken aan 'n rol.

F Die beskermingsmeganisme moet ontwikkel kan word om 'n hiërargie van sleutels te vorm, sodanig so dat verskillende vermoëns(sleutels) in die hiërargie vir verskillende individue gebruik kan word. Dit maak die beskermingsmeganisme soveel meer beheerbaar.

G 'n Uitgebreide of intensiewe *ouditmeganisme* bestaan, sodanig so dat enige verkeerdelike aksie onmiddellik uitgewys sal kan word. Dié meganisme verleen ook 'n beter mate van beheerbaarheid.

H Daar moet 'n sekerheidsbeampte wees, wat die hoogste *oorskryfreg* het tot die toegangsmeganisme.

I Die Stelselsekerheidsbeampte of die databasisadministreur moet die reg besit om toekenningsregte uit te deel vir die skep van nuwe entiteite.

1.3. DEFINISIES

Die volgende definisies sal geld vir die verloop van hierdie skrywe :

SEKERHEID

"The quality or state of being cost effectively protected from undue losses"[Lon87].

SEKERHEIDSVEREISTE

"The types and levels of protection necessary for equipment, data, information, applications and facilities"[Lon87].

SEKERHEIDSBELEID

"The set of laws, rules and practices that regulate how an organization manages, protects and distributes sensitive information"[Lon87].

"The statement of rules for one or more instances of communication. A security policy is based upon those services required and enforced by the appropriate system administration and also other security services requested by an entity wishing to communicate with the system"[Lon87].

SUBJEK

"In computer security, an active entity, generally in the form of a person, process or device that cause information to flow among objects or changes the system state"[Lon87].

OBJEK

"In computer security, a passive entity that contains or receives information. Access to an object potentially implies access to the information it contains. For example records, blocks, pages, segments, files, directories, directory trees and programs, as well as bits, bytes, words, fields, processors, etc." [Lon87].

ENTITEIT

"In databases, an object or event about which information is stored in a database"[Lon87].

VERMOë

"In computer security, an unforgeable ticket that is accepted by the system as incontestable proof that the presenter has authorized access to the object name by the ticket. It is often interpreted by the operating system and the hardware as an address for the object. Each capability also contains authorization information identifying the nature of the access mode"[Lon87].

ENKAPSULERING

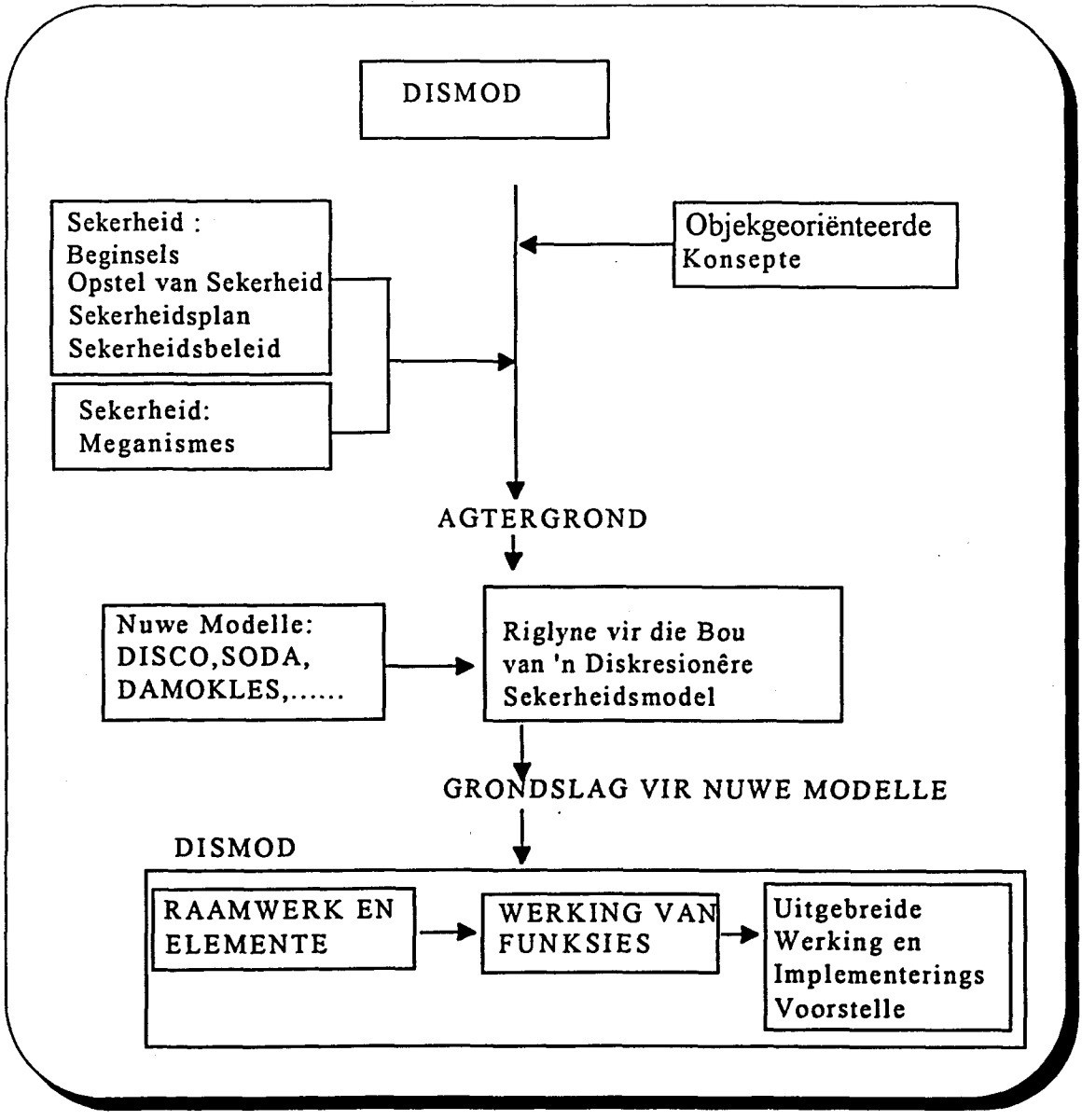
Enkapsulasie is wanneer 'n module (onafhanklike kode wat as klein, self-bevattende eenhede gebruik word) só funksioneer asof dit omring word deur 'n skerm wat enige onverwagte toegange van buite verhoed[Pfl89]. In 'n omgewing waar modules geënkapsuleerd is, vind wisselwerking slegs plaas deur middel van goed gedefinieerde koppelvlakke. 'n Module kan slegs gebruik word deur toegang op gespesifiseerde toegangspunte. 'n Module is in wisselwerking met die minste moontlike ander modules.

1.4. AFSLUITING

Die verloop van die verhandeling sal soos in figuur 1.1. gesien, as volg geskied:

- 1. Rekenaarsekerheid word in hoofstuk twee uiteengesit, waar 'n beskrywing gegee word van wat rekenaarsekerheid werklik is, hoe rekenaarsekerheid verkry kan word en watter

tipe evaluerings meganismes in rekenaarsekerheid bestaan. Hoofstuk drie is die afsluiting van rekenaarsekerheid en omvat die agtergrondskennis van rekenaarsekerheid, soos byvoorbeeld 'n bespreking van die onderskeie meganismes wat gebruik kan word om sekerheidstelsels en -modelle te implementeer.



FIGUUR 1 : UITEENSETTING VAN HOOFSTUKKE

- 2. Objekgeoriënteerde programeringsbeginsels en -konsepte speel ook 'n belangrike rol in die doelwitmodel en daarom word die grondbeginsels van die objekgeoriënteerde konsepte in hoofstuk vier behandel.
- 3. Noudat agtergrondskennis ingewin is oor sekerheid en objekgeoriënteerde konsepte is dit moontlik om 'n nuwe model te bou met behulp van hierdie beginsels. Hoofstuk 5 bied

nou die riglyne vir die bou van 'n objekgeoriënteerde sekerheidsmodel en let veral op huidige gebruike.

- 4. Hoofstuk ses, sewe en agt behandel die voorgestelde model, eerstens in breë trekke en daarna die volledige werking daarvan.
- Laastens word 'n gevolgtrekking gemaak oor die doeltreffendheid van die voorgestelde model.

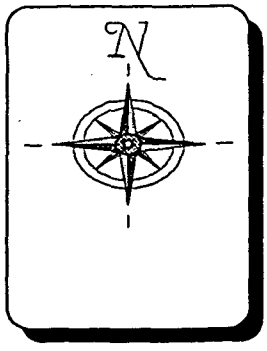
---oOo---

HOOFSTUK 2

ALGEMENE SEKERHEID

Die doel van hierdie studie, is om 'n nuwe sekerheidsmodel te bou wat die nuwe tegnologieë sal ondersteun. 'n Grondige kennis van algemene sekerheidsbeginsels en metodologieë is 'n noodsaaklikheid voordat 'n goeie model saamgestel kan word. Die doel van hoofstuk 2 is om 'n goeie basis van die grondbeginsels van rekenaarsekerheid daar te stel.

Die uiteensetting van hoofstuk 2 is soos volg:



Eerstens word beskryf wat sekerheid werklik is, en hoekom dit noodsaaklik is om sekerheid in elke organisasie te hê,
tweedens word daar gekyk na die bou van 'n sekerheidsmodel of die plan van aksie wat gevolg moet word om 'n veilige rekenaarstelsel te verkry,
en laastens word daar gekyk na die kwaliteite van 'n goeie rekenaarstelsel en die evalueringsmeganismes wat gebruik kan word om 'n sekerheidstelsel te klassifiseer volgens die kwaliteit van die stelsel.

2.1. INLEIDING

Misdade met betrekking tot rekenaars word gepleeg in al die verskillende bates van 'n rekenaarstelsel, onder andere in die apparatuur, programmatuur, die bergingsmedia, die data of ook deur persone in die organisasie wat gebruik maak van die rekenaarstelsel om hulle take uit te voer. In hierdie hoofstuk sal daar hoofsaaklik gelet word op sekerheid met betrekking tot die bates - programmatuur, apparatuur en

data omdat die ander tipes sekerheid wat uitgeoefen word, gewoonlik fisies van aard is.

Deur die verloop van hierdie gedeelte word aandag geskenk aan die volgende vrae:

"Wattter tipe bedreigings bestaan daar in enige rekenaaromgewing?"
"Hoe kan beskerming gebied word teen hierdie bedreigings?" en,
"Watter ander invloede bestaan daar in 'n rekenaaromgewing wat die sekerheidstelsel kan beïnvloed?."

In die beantwoording van die bogenoemde vrae word dit duidelik wat rekenaarsekerheid werklik is. Nadat daar 'n duidelike uiteensetting gegee is van wat rekenaarsekerheid werklik is, word die maniere waarop sekerheid in 'n rekenaaromgewing ingestel word, bespreek. Daarna word die sekerheidstelsel geëvalueer.

2.2. REKENAARSEKERHEID OORSPRONG EN OORSAKE.

Bedreigings in die rekenaaromgewing word hoofsaaklik gerig op die bates van die rekenaarstelsel, naamlik die data, apparatuur en programmatuur. Die volgende gedeelte is 'n beantwoordiging op die vraag - ***"Watter tipe bedreigings bestaan daar in die rekenaaromgewing?".***

Die volgende tipes bedreigings word deur oortreders gebruik in hul pogings om die nodige inligting te bekom óf te versteek, naamlik:

A. **Onderbreking** :

Onderbrekings kom voor in die bates van die rekenaarstelsel as dit verlore raak, nie meer beskikbaar is nie, of as dit in onbruik raak. Voorbeelde van onderbrekings is die skrapping van programme of datalêers of die faling van die bedryfstelselbestuurder.

B. **Onderskepping** :

Onderskepping is die bedreigings wat voorkom as 'n ongemagtigde party toegang verkry tot rekenaarbates. 'n Voorbeeld van só 'n ongemagtigde party is 'n persoon, 'n program of 'n rekenaarstelsel. 'n Voorbeeld van onderskepping is die ongemagtigde kopiëring van 'n program of datalêer, of die verkryging van inligting deur middel van lyntapping, ens.

C. **Verandering/Wysiging/Modifikasie**:

As die ongemagtigde party nie net toegang tot die data of rekenaarbates verkry nie, maar ook in staat is om dit te verander, word so 'n faling 'n verandering/wysiging of modifikasie genoem.

D. **Namaak/Fabrikasie**

'n Ongemagtigde party mag objekte van 'n rekenaarstelsel óf namaak óf afbeeld. Dié namaaksels kan dan by die bestaande stelsel gevoeg word en is dan 'n baie gevaarlike bedreiging vir die stelsel.

Die bogenoemde bedreigings is die grootste gevare waarteen 'n rekenaaromgewings beskerm moet word. Beskerming kan gebied word teen hierdie bedreigings, deur die volgende drie eienskappe in 'n sekerheidstelsel te implementeer, te gebruik en in stand te hou:

A. **Geheimhouding** :

Die afdwinging van geheimhouding in 'n sekerheidstelsel beteken dat die bates van die rekenaarstelsel só funksioneer dat dit ten alle tye slegs toegang toelaat vir gemagtigde partye tot die rekenaarbates.

B. **Integriteit** :

Integriteit in 'n rekenaar omgewing impliseer dat bates ten alle tye slegs deur gemagtigde partye verander kan word.

C. **Beskikbaarheid** :

Beskikbaarheid beteken dat die bates van die rekenaar beskikbaar gestel word aan gemagtigde partye.

Daar kan dus op hierdie stadium die gevolgtrekking gemaak word dat die uitoefening van die bedreigings in 'n rekenaaromgewing, verhoed kan word deur die implementering van die bogenoemde eienskappe in 'n rekenaaromgewing.
Die toepassing van die bogenoemde bedreigings op die bates van die rekenaarstelsel kan in figuur 2.1. gesien word.

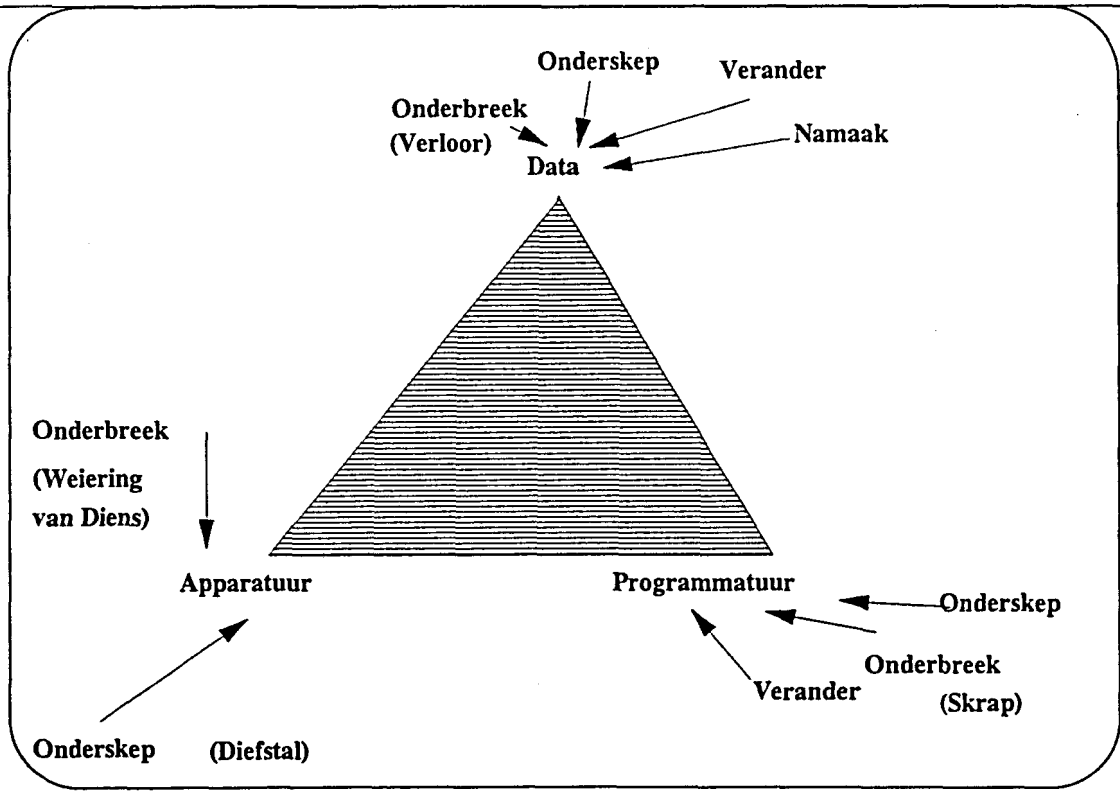


fig. 2.1. Beskerming van die bates van die rekenaarstelsel.

Die doel van rekenaarsekerheid is dus om geheimhouding, integriteit en beskikbaarheid van die rekenaarstelselkomponente te verseker.

Daar bestaan ook drie ander sekerheidsbeginsels wat die bou van 'n nuwe sekerheidstelsel kan beïnvloed, naamlik die volgende :

(a) Die Beginsel van die maklikste indringing.

Die beginsel van die maklikste indringing kan soos volg omskryf word: 'n Rekenaarindringer sal meestal die rekenaarstelsel probeer binnedring deur gebruik te maak van die maklikste aanvalwyse wat beskikbaar is. Die voorsorgmaatreël wat hierop getref moet word, is dat alle aspekte van die rekenaarstelsel beskou moet word en toetsing vir die maklikste moontlike manier van indringing gedoen moet word.

(b) Die beginsel van tydloosheid.

Die beginsel van tydloosheid kan soos volg weergegee word : Voorsorgmaatreëls moet getref word sodat 'n stelsel komponente so lank as wat dit waarde dra, genoeg beskerming moet geniet om sodoende die

indringing vir die indringer waardeloos te maak. Die voorsorgmaatreël verseker dat die inligting waardeloos is teen die tyd wat die indringer die inligting bekom. Dit wil sê die stelsel moet net lank genoeg beskerm word, sodat die inligting waardeloos is teen die tyd wat enigeen dit kan bekom.

(c) **Die beginsel van effektiwiteit.**

Dié beginsel stel voor dat die sekerheidstelsel bruikbaar moet wees en so gebruik moet word dat dit sy doel dien.

Deur gebruik te maak van die bogenoemde beginsels kan 'n meer effektiewe sekerheidstelsel gebou word.

Noudat dit duidelik is waarteen beskerming gebied moet word, asook wat gebruik moet word in die beskermingsproses, kan daar nou 'n volledige antwoord saamgestel word op die vraag van ***"Wat is sekerheid in rekenaarstelsels werklik?"***. Verskeie definisies bestaan, waaronder die volgende:

Sekerheid is die vermoë om inligting van waarde te beskerm, asook om die reg op individuele privaatheid in stand te hou[OSh91].

Sekerheid is die maatstaf van vertroue in die behoud van integriteit van beide data en die stelsel self.

Sekerheid is die versekering van die suiwerheid en die ononderbroke funksionering van stelsel wat 'n bruikbare diens voorsien.

(TCSEC) Enige stelling van rekenaarsekerheid begin met die daarstelling van 'n vereiste stelling, met ander woorde dit wat regtig bedoel word met 'n 'veilige' rekenaarstelsel. Oor die algemeen sal 'n veilige stelsel beheer uitoefen deur gebruik te maak van spesifieke rekenaarkenmerke, byvoorbeeld die uitoefening van toegangsbeheer deurdat slegs gemagtigde individue toegelaat word om die inligting te gebruik of deurdat slegs gemagtigde prosesse toegelaat word om verwerking te doen. 'n Voorbeeld hiervan word

gesien as slegs gemagtigde individue of prosesse byvoorbeeld toegelaat word om spesifieke objekte te lees, te skryf, by te werk of te skrap, maar slegs indien dit gemagtig is om daardie spesifieke funksies uit te voer.

(VSA Departement van Handel en Industrie)
Sekerheid is
(a) Die Geheimhouding van inligting of die verhindering van ongemagtigde blootstelling van inligting, asook
(b) Integriteit, d.w.s. die verhindering van ongemagtigde verandering aan inligting, of die ongemagtigde skrapping van inligting en laastens
(c) Beskikbaarheid of die verhindering van ongemagtigde weerhouding van inligting vanaf 'n bron.

Databasissekerheid is die beskerming van inligting wat in 'n databasis in stand gehou word[Fer81].

Dit is nou duidelik op hierdie stadium wat sekerheid is en watter beginsels deur 'n veilige stelsel ondersteun moet word om bedreigings die hoof te bied. Die vraag wat nou gevra kan word is :

"Hoe word rekenaarsekerheid verkry?"

Die antwoord op die bogenoemde vraag sal vervolgens beantwoord word deur die beskrywing van die oprigting van 'n rekenaarsekerheidstelsel. Daar sal in die volgende gedeelte veral verwys word na sekerheidsbeleide en verskeie modelle wat gebruik word in bestaande beleide.

2.3. VERKRYGING VAN REKENAARSEKERHEID.

Die daarstelling van rekenaarsekerheid volgens die bogenoemde definisies kan op verskeie maniere in 'n organisasie geïmplementeer word. In figuur 2.2. word 'n

voorstelling gegee van 'n metode wat gebruik kan word. Die metode word deur D.W. Roberts voorgestel[Rob90].

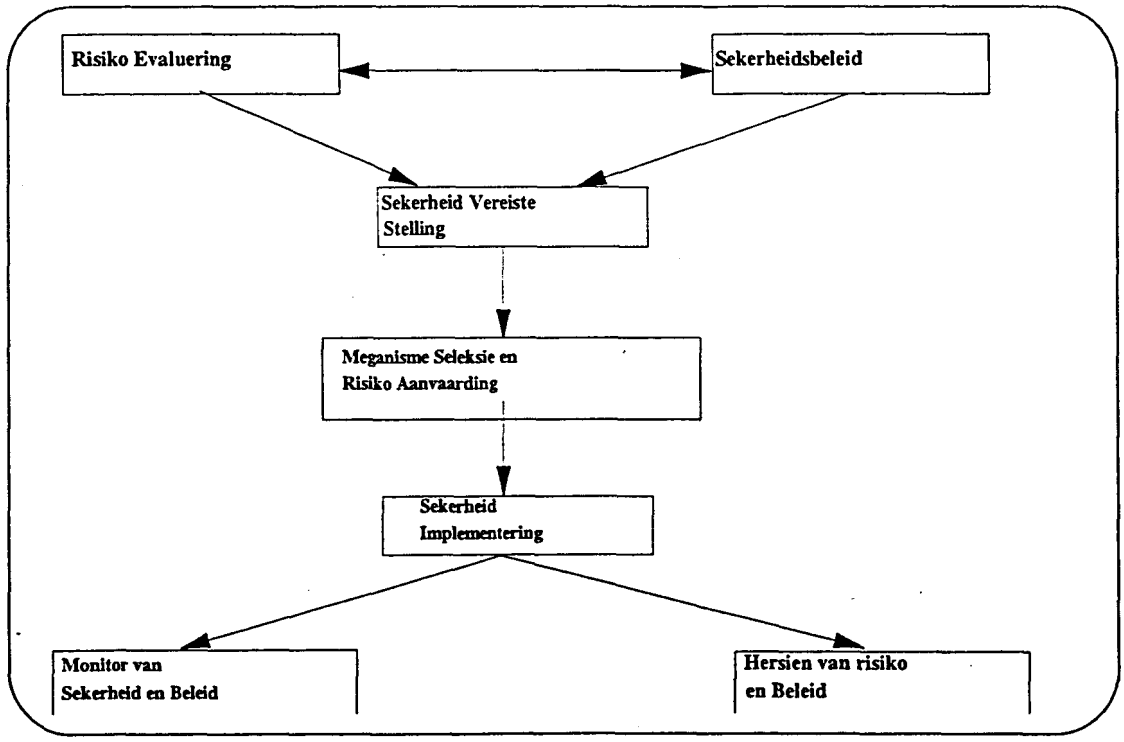


Fig. 2.2. Die Sewe stappe in die bereiking van sekerheid.

Die sewe stappe om sekerheid te verkry, soos gesien in figuur 2.2. is die volgende :

A. Risiko-evaluering

B. Die Stel van 'n sekerheidsplan

C. Die Stel van die sekerheidsvereistes van die organisasie

D. Meganismeseleksie en risiko-aanvaarding

Formele spesifisering van die sekerheidsmodel.

E. Sekerheidstelsel-implementering volgens die formele sekerheidsmodel.

F. Die Monitor van die sekerheid en afdwinging van die sekerheidsbeleid en laastens

G. Die Hersiening van die risiko-evaluering en die sekerheidsbeleid.

Elk van hierdie sewe stappe is ewe belangrik in die daarstelling van die sekerheidstelsel, maar slegs die opstel van die sekerheidsplan en die stel van die

sekerheidsvereistes van die organisasie sal in hoofstuk 2 bespreek word, terwyl die meganismeseleksie in hoofstuk 3 bespreek sal word.

2.3.1. DIE SEKERHEIDSPLAN.

Die sekerheidsplan is die hoeksteen van die organisasie se rekenaarsekerheid struktuur. 'n Sekerheidsplan is 'n stelling of verduideliking van die voorneme van die organisasie om beheer uit te oefen oor die toegang tot inligting en die verspreiding van inligting. 'n Kwaliteitsekerheidsplan word gebruik om die sekerheidstelsel te bou, daarom word die kwaliteite van die sekerheidstelsel alreeds in die sekerheidsplan uitgespel. Elke sekerheidsplan bevat dieselfde basiese beginsels en stipuleer ook 'n beleid vir sekerheid. Die sekerheidsplan is een van die moeilikste afdelings om goed te spesifiseer.

'n Sekerheidsplan identifiseer en organiseer die sekerheidsaktiwiteite van 'n rekenaarstelsel[Pfl89] en is 'n plan vir die huidige omstandigdhede sowel as toekomstige veranderinge wat mag plaasvind. 'n Sekerheidsplan moet byvoorbeeld van die volgende faktore melding maak, naamlik:

(A) Die **sekerheidsbeleid** :

'n Sekerheidsbeleid is die versameling van kriteria vir die voorsiening van sekerheidsdienste.

(B)**Huidige toestand** :

'n Beskrywing van die toestand van die sekerheid op die huidige oomblik.

(C) **Aanbevelings** :

Stappe wat sal lei tot die bereiking van die sekerheidsdoelwitte wat reeds geïdentifiseer is.

(D) **Verantwoordelikhede** :

Die besluit oor wie of wat verantwoordelik is vir die sekerheid van die stelsel

(E) **Tydtabel** :

'n Tabel wat aandui wanneer watter sekerheidsfunksies gedoen moet word.

(F) **Voortdurende Aandag**:

Die organisasie se verpligting tot sekerheid, naamlik 'n plan wat die struktuur vir 'n voortdurende hersiening van die sekerheidsplan spesifiseer.

Die sekerheidsbeleid vorm 'n baie belangrike gedeelte van die sekerheidsplan en word ook gebruik om die model waarvolgens die sekerheidstelsel gebou word, op te stel. Daar sal dus vervolgens eerstens gekyk word na die sekerheidsbeleid en verskeie sekerheidsmodelle en daarna na die evaluering van só 'n sekerheidsmodel.

2.3.2. DIE SEKERHEIDSBELEID.

Die sekerheidsbeleid vorm 'n baie belangrike gedeelte van die sekerheidsplan en kan ook die gehalte van die sekerheidsplan beïnvloed, daarom word daar spesiale aandag aan hierdie gedeelte gegee.

'n Sekerheidsbeleid is die versameling van kriteria vir die voorsiening van sekerheidsdienste. Dit kan reël-gebaseerd or identiteit-gebaseerd wees[Muf93].

'n Reëlgebaseerde sekerheidsbeleid is gebaseer op globale reëls vir alle gebruikers, wat vertrou op vergelykings wat gemaak word tussen die sensitiwiteit van die hulpbronne waartoe toegang vereis word, en die besit van die ooreenstemmende attribute van gebruikers, gebruikersgroepe of entiteite wat reageer in die belange van gebruikers.

'n Identiteit-gebaseerde sekerheidsbeleid is gebaseer op identiteite en/of attribute van gebruikers, gebruikersgroepe of entiteite wat reageer in

belange van die gebruikers, en die hulpbronne/objekte waartoe toegang vereis word.

'n Sekerheidsbeleid kan voorgestel word met behulp van 'n formele sekerheidsmodel om 'n hoë graad van presisie aan 'n sekerheidsmodel te verskaf[Muf93]. Voorbeelde van formele sekerheidsmodelle is die volgende:

Enkelvlakmodelle,

i. Monitormodelle, byvoorbeeld die verwysingsmonitor
ii. Inligtingsvloei-modelle

Traliemodel van Multi-vlak sekerheid

i. Die militêre sekerheidsmodel
ii. Die Traliemodel van toegangsekerheid

Inligtingsvloei-modelle

i. Bell-LaPadula-model
ii. Biba-model

Reël-gebaseerde modelle

i. Graham-Denning-model
ii. Harrison-Ruzzo-Ullman-model
iii. Take-Grant-stelsel

Elk van die bogenoemde modelle sal kortliks hanteer word om die verskil tussen 'n reël-gebaseerde en identiteit-gebaseerde sekerheidsbeleid uit te wys. Daar word gebruik gemaak van 'n identiteit-gebaseerde sekerheidsmodel in die bou van die nuwe voorgestelde model en dit is dus belangrik om hierdie gedeelte te verstaan.

'n Sekerheidsbeleid kan afgedwing word deur 'n formele sekerheidsmodel, byvoorbeeld die verwysingsmonitor, in 'n betroubare rekenbasis(trusted computing base)[OSh91] in te bou.

'n Verwysingsmonitor is 'n voorbeeld van 'n meganisme wat verantwoordelik is vir die afdwing van gemagtigde toegangsverwantskappe tussen subjekte en objekte in 'n stelsel. Die realisering van 'n verwysingsmonitor in 'n stelsel is die verwysingsvalideringsmeganisme. Die verwysingsvalideringsmeganisme moet die volgende vereistes bevat om aan die TCSEC-vereistes vir 'n veilige stelsel te voldoen[OSh91]:

(a) Dit moet peuterbestand wees,
(b) die meganisme moet altyd opgeroep word, en laastens
(c) die meganisme moet klein genoeg wees om geanaliseer en getoets te kan word sodat volledigheid daarvan verseker kan word.

'n Stelsel wat op die bogenoemde wyse, d.w.s. met enige tipe valideringsmeganisme, ontwerp en geïmplementeer word, implementeer in werklikheid 'n **sekerheidskern**. Valideringsmeganismes kan geïmplementeer word as deel van 'n algemene doelmeganisme, soos byvoorbeeld die bedryfstelsel. Die bedryfstelsel sal byvoorbeeld gebruik word, want dit bied die volgende sekerheidsdienste[Pfl89], nl:

(a) Geheuebeskerming
(b) Lêerbeskerming
(c) Algemene Objekbeskerming en
(d) Toegangsmagtiging of Toegangswaarmerking.

Die Betroubare Rekenbasis (BRB) word gebruik vir tipes stelsels waar valideringsmeganismes geïmplementeer word as deel van 'n algemene doelmeganisme van die algehele stelsel. 'n Betroubare Rekenbasis[OSh91] is

dié gedeelte van die stelsel wat al die elemente van die stelsel bevat wat verantwoordelik is vir die ondersteuning en afdwing van die sekerheidsbeleid. Dit wil sê daar kan ander meganismes, behalwe die sekerheidsmeganismes, in die Betroubare Rekenbasis bestaan.

Uit die bogenoemde kan afgelei word dat daar in die meeste sekerheidstelsels 'n meganisme bestaan wat gebruik word vir die afdwing van die sekerheid, asook 'n meganisme wat die veiligheid van die sekerheidsmeganismes beskerm. Daar sal vervolgens aandag geskenk word aan die verskillende tipes sekerheidsmodelle wat volg uit of deel vorm van die sekerheidsbeleid in 'n sekerheidsplan.

2.3.2.1. SEKERHEIDSMODELLE

Die sekerheidsmodel dien as voorstelling van die sekerheidsbeleid en speel 'n belangrike rol in die formulering van 'n effektiewe sekerheidstelsel en is ook die basis vir die toetsing van die effektiwiteit van die stelsel en moet dus duidelik uiteengesit word. Indien die sekerheidsmodel suksesvol beplan en ontwerp is, is die implementering van die sekerheidstelsel gebou op 'n vaste basis en is die moontlikhede vir 'n suksesvolle sekerheidstelsel soveel groter.

Die verskillende sekerheidsmodelle sal elk nou kortliks bespreek word. Daar word by die enkel vlak modelle begin.

DIE VERWYSINGSMODEL

'n Verwysingsmonitor of -model kan beskryf word as 'n hek tussen 'n subjek en 'n objek. Die monitor tree in werking met elke versoek en vir 'n spesifieke tipe toegang tot 'n objek wat gerig word deur subjekte. Die monitor reageer dan deur eerstens deur die toegangsbeheerinligting te konsulteer en daarna die toegang te weier of toe te laat afhangende van die resultaat van die navraag in

die toegangsbeheerinligting. 'n Figuurlike beskrywing van die verwysingsmonitor kan gesien word in figuur 2.3.

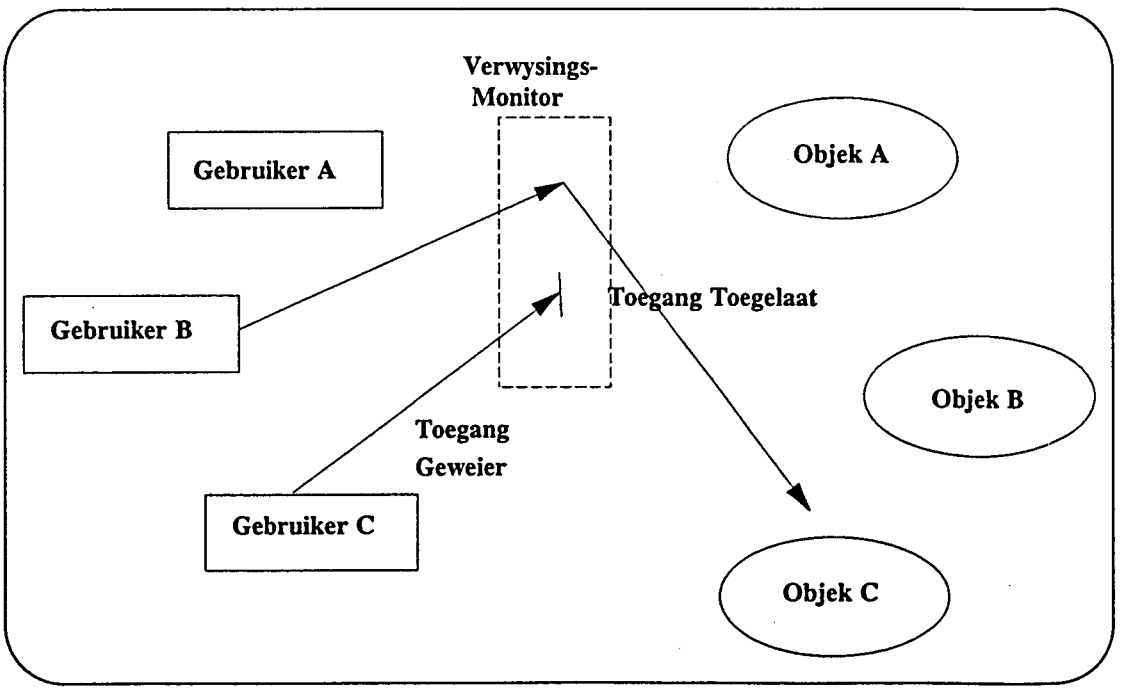


Fig 2.3. Die verwysingsmonitor.

Alhoewel die verwysingsmonitor die eenvoudigste sekerheidsmodel is, is dit nie die effektiefste nie om die volgende redes:

(a) Die feit dat alle gebruikersversoeke deur die monitor moet beweeg vir goedkeuring, bring dit mee dat daar bottelnekke kan voorkom in die tou van versoeke wat wag vir goedkeuring

(b) Slegs direkte toegang word beheer en nie indirekte inligtingsuitruiling nie.

DIE INLIGTINGSVLOEIMODEL

Die inligtingsvloeimodel is 'n oplossing vir die bottelnekprobleem (sien punt (a) in die bogenoemde model) in die verwysingsmodel. Die inligtingsvloeimodel

reageer as 'n intelligente filter met die oorplasing van inligting, indien toegang tot 'n spesifieke objek toegelaat is. Die inligtingsvloeimodel lyk soos volg:

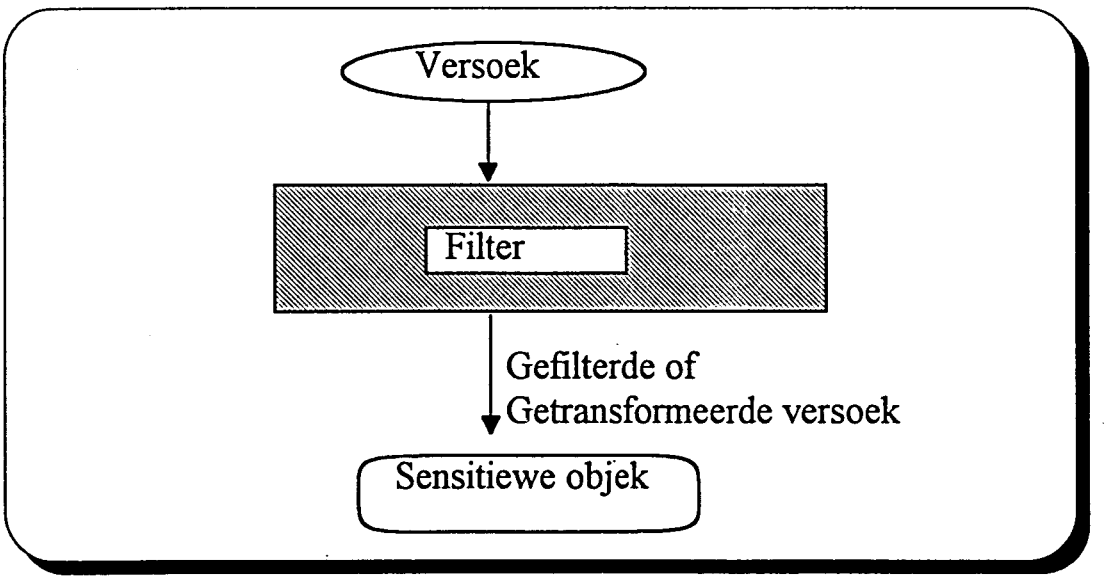


Fig 2.4 Die intelligente filter in die inligtingsvloeimodel.

Die inligtingsvloeimodel kan gebruik word om die potensiële toegange uit te wys terwyl die program nog vertaal word, d.w.s. voordat die program uitgevoer word. Die sekerheidsprosedure analiseer die vloei van inligting vir elke stelling in 'n program. Die analise kan bewys dat nie-sensitiewe uitvoere vanaf 'n module, nie op enige wyse geaffekteer word indien toegang geskied vanaf die module na sensitiewe data nie. Die bewys verifieer dat 'n gebruiker nie ongemagtigde data sal verkry as resultaat van die roep van 'n module nie. Die model is implementeerbaar op veldvlak.

TRALIE-MODEL VAN MILTIVLAKSEKERHEID.

Daar sal nou gevalle beskou word waar daar 'n reeks van grade van sensitiwiteit, beide vir objekte en gebruikers bestaan. Dié gevalle moet op só 'n wyse gemodelleer kan word dat daar in die model 'n duidelike beeld is van die gelyktydige hantering van gedeeltes inligting deur verskillende grade van sensitiwiteit. Die traliemodel is 'n voorbeeld van 'n veralgemening van die militêre model van inligtingsekerheid. Dié model staan bekend as die

traliemodel van sekerheid want sy elemente vorm 'n wiskundige struktuur , bekend as 'n tralie.

DIE MILITêRE SEKERHEIDSMODEL

In die militêre model word daar aan elke brokkie inligting 'n rang of belangrikheidsorde toegeken. Voorbeelde van sulke range is ongeklassifiseer, konfidensieel, geheim of hoogs geheim. Dié range wat toegeken word aan die inligtingsbrokkies is disjunk. Gebruikers maak gebruik van dié sensitiewe data om hul werk te doen.

Een sekerheidsbeginsel wat voorheen geïdentifiseer is, naamlik die **beginsel van die minste voorreg,** spesifiseer dat 'n subjek slegs toegang moet besit tot die minste moontlike objekte wat nodig is om hom in staat te stel om suksesvol te kan werk, word gebruik in die militêre sekerheidsmodel. Toegang tot inligting word dus beperk deur die **"Nodig-om-te-weet"-reël,** wat toegang tot sensitiewe data net verleen aan subjekte wat dit nodig het om die data te gebruik in die uitvoering van hul take.

In hierdie model word elke brokkie geklassifiseerde inligting geassosieer met een of meer projekte, bekend as **kompartemente,** wat die subjek-inhoud van die inligting beskryf. Kompartemente word gebruik om die nodig-om-te-weet-beperkings af te dwing sodat die mense slegs toegang kan verkry tot inligting waarvan die inhoud relevant vir hul werk is. 'n Enkele brokkie inligting word gekodeer in een, twee of meer kompartemente, afhangende van die kategorieë waaraan dit verwant is. Die kombinasie <rang,kompartement> word 'n klas of **klassifikasie** van 'n gedeelte van inligting genoem.

'n Persoon wat toegang tot sensitiewe inligting wil hê, moet 'n klaring besit vir die inligting. 'n Klaring is die aanduiding wat 'n persoon het om toegang tot

inligting tot op 'n sekere vlak van sensitiwiteit te kry of aandui dat die persoon sekere kategorieë sensitiewe inligting mag sien. Die klaring van 'n subjek is 'n kombinasie <rang, kompartemente>. Die kombinasie het dieselfde vorm as die klassifikasie van 'n gedeelte inligting.

Daar bestaan ook die volgende verwantskap wat moet geld in hierdie model, naamlik dat:

O $\leq$ S as en slegs as , waar O 'n objek en S 'n subjek is
rang o $\leq$ rang s en (rang = klaringvlak)
kompartement o kompartement S.

Die relasie $\leq$ in die bogenoemde vergelyking word gebruik om die sensitiwiteit, asook die inhoud van die inligting waartoe 'n subjek toegang besit, te beperk. 'n Subjek kan dus toegang kry tot 'n objek as en slegs as die klaringsvlak van die subjek ten minste so hoog is soos dié van die inligting, en die subjek die nodig-om-te-weet-toegang tot alle kompartemente waarvoor die inligting geklassifiseer is, het. 'n Voorbeeld van die bogenoemde is die volgende: Veronderstel dat daar vier tipe klarings bestaan: ongeklassifiseerd, konfidensieel, geheim en hoogs geheim (genoem in die orde van belangrikheid). As inligting dan geklassifiseer word as <geheim,personeel_salarisse>, kan dit slegs gebruik word deur subjekte wat 'n klaring <hoogs geheim,personeel_salarisse> of <geheim,personeel_salarisse> het. Dit kan nie deur 'n subjek met 'n klaring van <konfidensieel,personeel_salarisse> gebruik word nie.

Die militêre sekerheidsmodel dwing beide sensitiwiteitsvereistes en die nodig-om-te-weet-vereistes af. Sensitiwiteitsvereistes is hiërargiese vereistes terwyl nodig-om-te-weet beperkings nie-hiërargies van aard is.

TRALIE MODEL VAN TOEGANGSEKERHEID.

'n Tralie is 'n wiskundige struktuur van elemente wat onderwerp word aan 'n relasionele operator. Die elemente van 'n tralie is georden volgens 'n gedeeltelike ordening . 'n Gedeeltelike ordening is 'n relasie wat transitief en antisimmetries is wat beteken dat vir elke drie elemente a,b,c :

transitief : as $a \leq b$ en $b \leq c$ dan is $a \leq c$,

antisimmetries : as $a \leq b$ en $b \leq a$ dan is $a = b$.

Pare elemente is nie altyd vergelykbaar in 'n tralie nie, maar enige twee elemente het 'n grootste bogrens en 'n kleinste ondergrens. Die militêre sekerheidsmodel is 'n voorbeeld van 'n tralie. 'n Sekerheidstelsel wat ontwerp is om 'n traliemodel te ondersteun,kan gebruik word in 'n militêre omgewing. Dit kan ook gebruik word in kommersiële omgewings met ander etikette vir die sekerheidsgraderings.

INLIGTINGSVLOEIMODELLE

BELL-LAPADULA-MODEL

(implementeer geheimhouding)

Die Bell-LaPadula-model[Pfl89,OSh91] is 'n formele beskrywing van die toelaatbare paaie van inligtingsvloei in 'n veilige stelsel. Die doelwit van die model is om toelaatbare kommunikasie te identifiseer waar dit belangrik is om geheimhouding in stand te hou. Die model was gebruik om sekerheidsvereistes te definieer vir stelsels wat gelyktydig data op verskillende sensitiwiteitsvlakke hanteer. Beskou 'n sekerheidstelsel met die volgende eienskappe. Die stelsel omvat 'n versameling subjekte S en 'n versameling objekte O. Vir elke subjek s in S en elke objek o in O is daar 'n vaste sekerheidsklas C(s) en C(o). Die sekerheidsklasse word georden volgens 'n relasie. Twee eienskappe karakteriseer 'n geheime vloei van inligting :

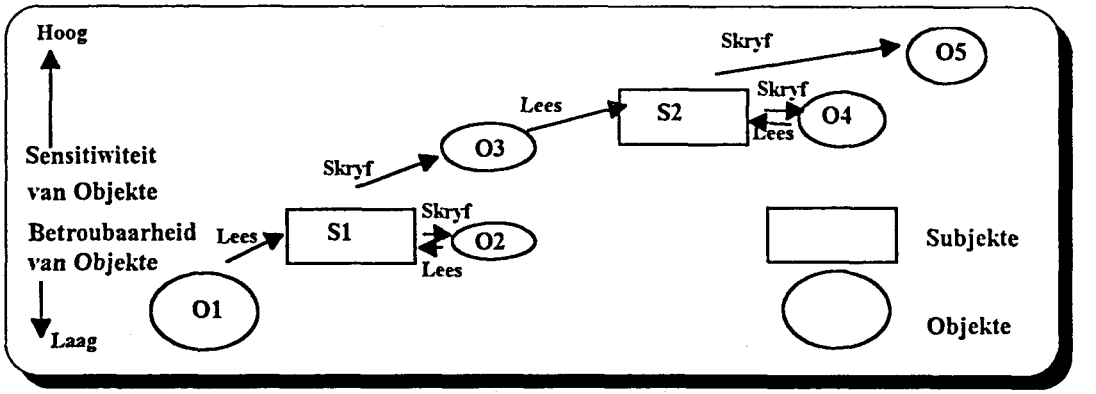


Fig 2.5. Die Bell-en-Lapadula-model

Die Eenvoudige Sekerheidseienskap :

'n Subjek s mag leestoegang besit tot 'n objek o as en slegs as

$C(o) \leq C(s)$

wat beteken dat die sekerheidsklas van iemand wat inligting ontvang ten minste so hoog moet wees as die klas van die inligting. Sien fig 2.5. vir die uiteensetting.

Die *-Eienskap

'n Subjek s met toegang tot 'n objek o mag ook skryfregte tot objek p besit as en slegs as : $C(o) \leq C(p)$. Sien fig 2.5. vir die uiteensetting. Die *-eienskap vereis dat 'n persoon wat inligting ontvang op een vlak nie moet kan kommunikeer met subjekte wat klaring besit op vlakke laer as die vlak van die inligting nie.

Let wel :

Die verwysingsmonitormodel en die Bell-LaPadula-model vorm die basis van die VSA Dept van Verdediging van die VSA se Betroubare Rekenaarsekerheidsevaluasiestandaard (TCSEC).

'n Soortgelyke inligtingsvloeimodel as die Bell-en-Lapadula sekerheidsmodel is die Biba-model, wat vervolgens hanteer sal word.

BIBA-MODEL

(Implementeer integriteit)

Biba[Pfl89] se model verhoed ongemagtigde verandering van data. Biba definieer integriteitsvlakke, in plaas van klaringsvlakke. Subjekte en objekte word georden volgens 'n integriteitsklassifikasieskema, aangedui deur I(s) en I(o), oftewel die integriteitsklasse van subjekte en objekte. Die eienskappe is :

Eenvoudige integriteitseienskap.

As subjek s vir objek o kan verander, dan is $I(s) \geq I(o)$

Integriteit *-Eienskap

As subjek s vir objek o kan lees met integriteitsvlak I(o) dan het s skryf toegang tot 'n objek p as en slegs as $I(o) \geq I(p)$.

Teoretiese Beperkings van sekerheidstelsels

GRAHAM-DENNINGMODEL.

Die Graham-Denningmodel [Pfl89]stel die konsep van 'n formele stelsel van beskermingsreëls voor. Graham en Denning het 'n model gekonstrueer met generiese beskermingseienskappe.

Die model funksioneer op 'n versameling subjekte S, 'n versameling objekte O, en 'n versameling regte R, en 'n toegangsbeheer matriks A. Die matriks het een ry vir elke subjek en een kolom vir elke subjek en elke objek. Die regte van 'n subjek op 'n ander subjek of 'n objek word getoon in die inhoud van 'n element van die matriks. Vir elke objek word daar aan een subjek spesiale regte toegeken en hy staan bekend as die **eienaar.** Vir elke subjek word 'n ander subjek met spesiale regte toegeken, en hy staan bekend as die **kontroleerder**.

In die Graham-Denningmodel is daar agt primitiewe beskermingsregte. Dié regte word gefraseer as bevele wat uitgereik kan word deur subjekte met effekte op ander subjekte of objekte. Hulle is die volgende :

Skep-objek : laat die subjek toe om 'n nuwe objek in die stelsel te skep,
Skep-objek, skrap-objek, skrap-subjek : Sien bogenoemde definisie
Leestoegangsreg: laat subjek toe om die huidige toegangsreg van 'n subjek tot 'n objek te bepaal.
Toekenning van toegangsreg: laat die eienaar van 'n objek toe om 'n toegangsreg vir 'n objek aan 'n ander subjek toe te ken
Skrapping van toegangsreg; laat 'n subjek toe om 'n reg vir 'n objek van 'n ander subjek te skrap op voorwaarde dat die subjek die eienaar van die objek is of dat die subjek die ander subjek waarvan die reg geskrap moet word, kontroleer.
Oorplaas van toegangsreg : laat 'n subjek toe om een van sy regte vir 'n objek oor te plaas na 'n ander subjek.

Dié stel reëls voorsien die eienskappe wat nodig is om 'n toegangsbeheermeganisme van 'n beskerming stelsel te modelleer. Die meganisme kan byvoorbeeld 'n verwysingsmonitor of 'n stelsel van deling tussen twee onbetroubare substelsels verteenwoordig.

HARRISON-RUZZO-ULLMANMODEL.

Die Harrison-Ruzzo-Ullmanmodel is ook gebaseer op bevele, waar elke bevel kondisies en primitiewe bewerkings gebruik.
Die struktuur van die bevele is soos volg :

Bevel naam(o_1,o_2,....,o_k)

as r_1 in A[s_1,o_1] en

r_2 n A[s_2,o_2] en

...

r_m in A[s_m,o_m]

dan

op_1.

op_2

....

op_n

end

Die bevel het die struktuur van 'n prosedure met parameters $o_1..o_k$.

In die model is elke subjek ook 'n objek. D.w.s die kolomme van die toegangsbeheermatriks is alle subjekte en alle objekte wat nie subjekte is nie. Al die parameters word hierteenoor gemerk as o, alhoewel hulle of subjekte, of nie-subjek-objekte kan bevat. Elke **r** is 'n generiese reg, en elk **op** is 'n primitiewe operasie. Die volgende operasies (op's) word in die model gebruik, naamlik:

skep subjek s

skrap objek o

vernietig subjek s

vernietig objek o

Invoer van reg in matriks posisie A[s,o]

Skrap van reg r uit matriks posisie S[s,o]

TAKE-GRANT-STELSELS.

In die Take-Grant-model[Pfl89,OSh91] is daar slegs vier primitiewe operasies, naamlik skep, wegneem(revoke), neem(take) en toeken(grant). Wegneem en skep is soortgelyk aan die wegneem- en skep-operasies as in die Graham-Denningmodel.

Beskou 'n stelsel waar S die versameling subjekte, O die versameling objekte is en waar objekte óf aktief óf passief is. Laat R 'n versameling regte wees. Elke subjek of objek word aangedui deur 'n nodus van 'n grafiek. Die regte van 'n spesifieke subjek op 'n spesifieke objek word aangedui deur 'n gemerkte gerigte lyn van subjek na die objek.

Laat s die subjek wees wat al die operasies uitvoer. Die vier operasies word soos volg gedefinieer :

Skep(o,r) : 'n Nuwe nodus met etiket o word by die grafiek gevoeg. Vanaf s na o is daar 'n gerigte lyn met etiket r getrek, wat die regte van s op o aandui.

Wegneem(o,r): Die versameling regte r van s op o word weggeneem. Veronderstel die lyn van s na o was geëtiketeer q (verenig) met r; na die wegneembevel(o,r) word die etiket vervang met slegs q. S kan informeel sy regte "revoke" om net reg r op o te doen.

Toeken(o,p,r): Subjek s sal dan toegangsregte r op p toeken vir o. Subjek s kan toegangsreg r op p vir o toeken as en slegs as s 'n toeken reg besit op o en s "- r-regte" op p besit. S kan inteendeel enige van sy regte met o deel so lank as wat s die reg het om voorregte uit te deel(toe te ken) op o.

Neem(o,p,r): S neem die toegangsreg r op p weg van s af. Subjek s kan net die toegangsreg r op p van o af wegneem as s die "take"-reg op o het, en o het 'n "r reg" op p. D.w.s. s kan enige regte van o af wegneem as s net die reg van wegneemvoorregte op o het.

Dit was kortliks die uitleg van die Harrison-Ruzzo-Ullman-model, en ook die afsluiting van sekerheidsmodelle.

Dit is nou duidelik wat 'n sekerheidsbeleid is en hoe dit geïmplementeer kan word. Daar sal vervolgens aandag geskenk word aan die evaluering van die sekerheidsplan.

2.4. EVALUERING VAN SEKERHEIDSMODELLE.

Evaluering van 'n formele sekerheidmodel word gebruik om die graad van veiligheid van die sekerheidsmodel te bepaal en om vas te stel of daar aan alle vereistes vir die stelsel voldoen word. Daar bestaan verskeie evalueringsmeganismes wat gebruik word om 'n sekerheidstelsel te evalueer, waaronder die volgende : TCSEC, OSI en ITSEC. Die doelwit van hierdie evalueringsmeganismes is om die veiligheid van sekerheidmodel te klassifiseer in verskillende kategorieë[OSh91]. Die evalueringsmeganismes kan ook gebruik word as riglyne in die opstel van die sekerheidstelsel.

Die vereistes van die verskillende evalueringsmeganismes sal nou beskou word vir die volgende redes :

(a) Dit gee 'n aanduiding van watter eienskappe 'n goeie sekerheidsmodel veronderstel is om te besit,

(b) Dit sal die leser in staat te stel om die model wat in hierdie skrywe gebou word, te kan evalueer.

Die evalueringsmeganismes TCSEC, OSI en ITSEC sal vervolgens bespreek word.

TCSEC

Die VSA se departement van Verdediging het ses vereistes vir veilige rekenaarstelsels geïdentifiseer, waaruit die definiëring van die vereistes vir 'n veilige rekenaarstelsel in die Oranjeboek gevolg het.

Die ses vereistes[Pfl89,Muf93] vir 'n kwaliteit sekerheidstelsel is die volgende:

A. Die **Sekerheidsbeleid of Sekerheidsplan**

Daar moet 'n duidelik en goed gedefinieerde sekerheidsbeleid bestaan wat afgedwing word deur die sekerheidstelsel.

B. **Identifikasie.**

Elke subjek moet uniek en oortuigend geïdentifiseer kan word. Identifikasie is nodig sodat subjek-/objek-toegangsversoeke getoets kan word vir geldigheid.

C. **Etikettering**

Elke objek moet geassosieer word met 'n etiket wat die sekerheidsvlak van die objek aandui. Die assosiasie, wat bekend staan as etikettering van die objek, moet gedoen word, sodat die etiket te enige tyd beskikbaar is vir vergelyking as toegang na die objek vereis word.

D. **Verantwoordelikheid**

Die stelsel moet volledig rekord hou van aksies wat sekerheid affekteer en hierdie rekords moet in veilige bewaring wees. Die tipe vaslegging van aksies staan bekend as 'n **ouditspoor**. Die ouditspoor sluit aksies soos die bekendstelling van nuwe gebruikers aan die stelsel, die toekenning/verandering van die sekerheidsvlak van 'n subjek/objek en geweierde toegangspogings, in.

E. **Versekering**

Die apparatuur en programmatuur wat doeltreffende veiligheid bewerkstellig moet ter enige tyd geëvalueer kan word.

F. Die apparatuur wat die sekerheidstelsel bevat en die programmatuur wat sekerheid afdwing moet deurlopend beskerm word teen ongemagtigde veranderinge.

TCSEC verdeel sekerheidstelsels volgens die graad van veiligheid wat aangebied word in die volgende 4 kategorieë, naamlik:

D - Minimale Beskerming,

C - Diskresionêre Beskerming,

B - Verpligte Beskerming en

A - Geverifiëerde Beskerming.

Die volgende tabel[Pfl89] lys die vereistes vir elke kategorie :

Kriteria	Vereites						
	D	C1	C2	B1	B2	B3	A1
Sekerheidsbeleid :							
Diskresionêre Toegangsbeheer	N	A	A	S	S	A	S
Objek-hergebruik	N	N	A	S	S	S	S
Etikette	N	N	N	A	A	S	S
Etiket-Integriteit	N	N	N	A	S	S	S
Uitvoer van gemerkte inligting	N	N	N	A	S	S	S
Merk van Menslikleesbare uitvoer	N	N	N	A	S	S	S
Verpligte Toegangsbeheer	N	N	N	A	A	S	S
Subjeksensitiwiteitsmerkers	N	N	N	N	A	S	S
Toestel Merkers	N	N	N	N	A	S	S
Verantwoordelikheid :							
Identifisering en Sertifisering	N	A	A	A	S	S	S
Oudit	N	N	A	A	A	A	S
Betroubare Pad	N	N	N	N	A	A	S
Versekering :							
Stelselargitektuur	N	A	A	A	A	A	S
Stelsel-integriteit	N	A	S	S	S	S	S
Sekerheidstoetsing	N	A	A	A	A	A	A
Ontwerp Spesifikasie en Verifikasie	N	N	N	A	A	A	A
Geheime Kanaalanalise	N	N	N	N	A	A	A
Betroubare Fasiliteitsbeheer	N	N	N	N	A	A	S
Konfigurasiebeheer	N	N	N	N	A	S	A
Betroubare Herstel	N	N	N	N	N	A	S
Betroubare Verspreiding	N	N	N	N	N	N	A
Dokumentasie :							
Sekerheidskenmerk-gebruikersgids	N	A	S	S	S	S	S
Betroubare Fasiliteitsgids							
Toets Dokumentasie	N	A	A	A	A	A	S
Ontwerp Dokumentasie	N	A	S	S	A	S	A
	N	A	S	A	A	A	A

N - Geen vereistes, A - Addisionele vereistes, en S - Dieselfde Vereistes as die vorige klas.

Tabel 2.1. Vereistes vir elke TCSEC-veiligheidskategorie.

Elk van die bogenoemde kategorieë sal nou kortliks bespreek word. Uit Tabel 2.1. kan die kriteria vir die verskillende evalueringsklasse soos volg saamgevat word :

Klas D is die minimale beskermingsklas en word gereserveer vir stelsels wat faal om te voldoen aan die evalueringskriteria vir die hoër klasse.

Klas C1 word die diskresionêre sekerheidsbeskermingsklas genoem. Dié klas is bedoel as 'n omgewing waar daar gesamentlike gebruik en verwerking van data met dieselfde sensitiwiteitsvlak is. Op hierdie sekerheidsvlak moet daar 'n skeiding voorsien word tussen die gebruikers en die data. Daar moet ook genoegsame toegangsbeheermaatreëls bestaan om toe te laat dat gebruikers hul eie data beskerm. Die Betroubare rekenbasis moet gebruikersidentifisering met waarmerking deur 'n beskermde meganisme soos 'n wagwoordmeganisme afdwing[OSh91]. Die stelselargitektuur moet 'n domein vir die Betroubare Rekenbasis in stand hou,sodat dit beskerm is teen ongemagtigde verandering van sy kode of data[OSh91]. ('n Stelsel moet 'n domein besit om 'n C1-klassifisering te kan besit.[Pfl89]. Dit sluit sekerheidsfunksies in, asook peuter bestandheid. 'n Toetsvereiste vir 'n C1-klassifisering is dat die stelsel moet werk soos gespesifiseer in die stelseldokumentasie en dat daar geen sigbare of duidelike wyse bestaan waarop die stelsel gepenetreer kan word nie. 'n Voorbeeld van 'n stelsel met 'n C1-klassifisering is die IBM MVS-bedryfstelsel[Pfl89] en RACF(Resource Access Control Facility)[Pfl89].

Klas C2 word gekenmerk deur 'n fyner grein van diskresionêre toegangsbeheer kontroles. Beskerming moet implementeerbaar wees op die vlak van die enkel gebruiker. D.w.s., individuele betroubaarheid deur aantekenprosedures, beskermde ouditrekords van sekerheidsverwante aksies en isolering van sekerheidsverwante objekte deur die Betroubare Rekenbasis. Die Betroubare Rekenbasis moet ook verseker dat herbruikbare objekte nie residue data (data wat agterbly op primêre en sekondêre geheue of register na 'n proses beëindig is) bevat indien dit toegeken word nie. Skyfblokke moet byvoorbeeld nie data bevat van 'n

vorige geskrapte leêr nie. 'n Voorbeeld van 'n klas C2 geklassifiseerde stelsel is die IBM MVS-bedryfstelsel[Pfl89] met die bygevoegde pakket ACF2, sowel as die VAX bedryfstelsel MVS.

Klas B1 (Geëtiketteerde sekerheidsbeskerming). Vanaf vlak B1 hoër boontoe sluit alle klasse verder 'n verpligte sekerheidstoegangbeheermeganisme in. Op hierdie klassifiseringsvlak moet elke objek gemerk word met 'n sekerheidsvlak, en hierdie etikette moet gebruik word as die basis vir toegangsbeheerbesluite. Die toegangsbeheer moet gebaseer word op 'n model wat beide gebruik maak van hiërargiese en nie-hiërargiese vlak kategorieë. Die verpligte toegangsbeleid is gebaseer op die Bell-LaPadula-model. Dit wil sê dat 'n B1-stelsel 'n Bell-Lapadula-beheer moet implementeer vir alle toegange, en dan kan gebruiker-diskresionêre toegangsbeheerkontrole verdere limiete op toegang plaas. Ontwerpsdokumentasie, bronkode en objekkode word onderwerp aan deeglike analise en toetsing.

Klas B2 (Gestruktureerde Beskerming). 'n Verdere verbetering op die vorige klasse is dat hierdie klas 'n deeglike toetsing en hersiening moet ondergaan. 'n Verifieerbare top-vlak ontwerp moet voorgestel word en die toetsing moet bevestig dat die stelsel die ontwerp wel implementeer. Die stelsel moet intern gestruktureer word in 'n "goed-gedefinieerde, grootliks onafhanklike modules" . Die beginsel van die minste voorreg moet afgedwing word in hierdie ontwerp. Toegangsbeheerbeleide moet afgedwing word op alle objekte en subjekte, insluitende toestelle. Analise van geheime kanale word vereis. 'n Gebruiker word uitgewys as die sekerheidsbeampte: die gebruiker sal die toegangsbeheer beleid implementeer, terwyl die operateur slegs funksies uitvoer wat verwant is aan die kontinue werking van die stelsel. Die beskermingsstelsel moet 'n beskermingsdomein in stand hou vir sy eie uitvoering, die domein moet die integriteit van die stelsel verseker teen eksterne onderbrekings of peuterings. 'n Voorbeeld van 'n Klas B2 bedryfstelsel is MULTICS[Pfl89].

Klas B3 (Sekerheidsdomeine) Die sekerheidsfunksies van 'n klas B3-stelsel moet klein genoeg wees vir intensiewe toetsing. Die hoë-vlak ontwerp moet volledig en eenvoudig in opvatting wees, en 'n "oortuigende argument" moet bestaan dat die stelsel die ontwerp implementeer. Die implementering van die ontwerp sal beduidende gebruik van lae, abstraksie- en inligtingsversteking bevestig. Subjek-/objekdomeine word vereis met 'n vermoë om toegangsbeskerming vir elke objek te implementeer, met die aanduiding van toegelate subjekte, tipes toegang wat toegelaat word vir elk en geweierde subjekte. Die volle verwysingsmonitor-konsep sal geïmplementeer word, sodat elke toegang getoets word. Die sekerheidsfunksies moet peuterbestand wees. Die stelsel moet ook verder baie weerstand kan bied teen penetrasie.

Klas A1 (Geverifieerde Ontwerp). Die klas van sekerheidstelsels vereis 'n formeel geverifieerde stelselontwerp. Die vermoëns van die stelsel is dieselfde as vir klas B3. Daar is vyf belangrike kriteria vir klas A1 sertifisering, nl:

(1) 'n Formele model van die beskermingstelsel en 'n bewys van sy konsekwentheid en genoegsaamheid

(2) 'n Formele top-vlakspesifikasie van die beskermingstelsel,

(3) 'n Demonstrasie dat die top-vlakspesifikasie ooreenstem met die model

(4) 'n Informele implementering waar daar aangetoon moet word dat dit konsekwent is met die spesifikasies

(5) 'n Formele analise van geheime kanale. Die Honeywell Scomb-stelsel [Pfl89] word as 'n A1-graderingsekerheidsteslsel beskou.

Die bogenoemde gedeelte omvat grootliks die TCSEC evalueringsmeganisme en soos reeds voorheen genoem, is dit gebaseer op 'n samestelling van die verwysingsmonitor en die Bell-en-Lapadula-model in 'n Betroubare Rekenbasis.

OSI - Die Oopstelsel-interskakeling

Die OSI sekerheid evalueringsmeganisme is ontwerp vir Oop-interskakelende stelsels. Interskakelende stelsels deel baie van die verantwoordelikheid vir sekerheid tussen die verskillende stelsels wat gesamentlik funksioneer, maar aandag word veral geskenk aan die vatbaarheid van die kommunikasieskakels tussen die verskillende stelsels[OSh91].

Die TCSEC-byvoegings vir 'n Oop stelsel interskakelende omgewing omvat veertien sekerheidsdienste wat in die omgewing geïmplementeer moet word, om die omgewing veilig te maak. Dié dienste word met verskeie van die vlakke van die OSI verwysingsmodel geassosieer.

Die sekerheidsdienste wat deur 'n OSI omgewing geïmplementeer moet word, is die volgende :

(a) **Eweknie-entiteitwaarmerking** :

Die sekerheidsdiens verseker die korrektheid van die identiteit van 'n afgeleë party.

(b) **Data-oorsprongwaarmerking**:

Die tipe waarmerking verseker dat die data wel gekom het vanaf die beweerde bron.

(c) **Toegangsbeheerdienste** :

Dié tipe dienste voorsien toegangsbeheer op alle netwerk objekte, insluitende sessies wat aangevra word vanaf afgeleë entiteite.

(d) **Verbindingskonfidensialiteit**:

Konfidensialiteit van al die data in 'n kommunikasiesessie tussen twee entiteite word verseker deur hierdie diens.

(e) **Verbindinglose konfidensialiteit**:

Konfidensialiteit van data gestuur tussen partye wat nie in 'n kommunikasiesessie verbind is nie, word verseker. Die partye is byvoorbeeld nie in 'n sessie verbind nie omdat die protokolle wat hulle gebruik nie verbinding-georiënteerde sessies ondersteun nie.

(f) **Selektiewe veldkonfidensialiteit**:

Die sekerheidsdiens verseker die konfidensialiteit van data in geselekteerde velde tydens die kommunikasieproses, byvoorbeeld die oorplasing van data.

(g) **Verkeersvloei-konfidensialiteit**:

Die blootstelling van data deur byvoorbeeld die ontleding van netwerk laaiing en boodskaproetering word deur hierdie diens verhoed.

(h) **Verbindingsintegriteit met die moontlikheid om te herstel**:

Dié sekerheidsdiens verseker dat data korrek oorgeplaas word, dit wil sê, in die korrekte volgorde en sonder enige veranderinge. Die diens sal 'n poging aanwend om herstel na die oorspronklike situasie te bewerkstellig in die geval waar probleme ontdek word.

(i) **Verbindingsintegriteit sonder die moontlikheid om te herstel**:

Die tipe diens is dieselfde as (h), maar die moontlikheid om te herstel word nie aangebied nie.

(j) **Selektiewe veldverbindinglose herstel:**

Die tipe diens is ook dieselfde as (h), maar is van toepassing op geselekteerde velde.

(k) **Geen ontkenning van oorsprong** :

Die diens verseker dat die sender van 'n boodskap nie in staat is om te ontken dat hy die boodskap gestuur het nie, deur byvoorbeeld data te inkorporeer in die boodskap, wat slegs deur die sender geproduseer kan word.

(l) **Geen ontkenning van ontvangs**:

Die diens verseker dat die ontvanger van 'n boodskap nie in staat is om te ontken dat hy die boodskap ontvang het nie, deur byvoorbeeld 'n erkenning van ontvangs só te gebruik dat die identiteit van die ontvanger duidelik is.

Die bogenoemde dienste kan geïmplementeer word deur byvoorbeeld gebruik te maak van enkripsie, digitale handtekeninge, toegangsbeheermeganismes, data-integriteit of foutopsporingsmeganismes, waarmerkuitruitling, verkeers-vulling(oortolligheid in boodskappe), roeteringsbeheer, ens.

2.5. NETWERK TCSEC (TNI)

Die NETWERK-interpretasie van TCSEC[OSh91] identifiseer twee alternatiewe sienings wat van toepassing is op stelsels met netwerke, nl. :

(A) DIE ENKELBETROUBARE STELSELSIENING EN

(B) DIE INTERSKAKELENDE ERKENDE SIENING.

Dié twee sienings sal vervolgens in meer besonderhede bespreek word.

Addisionele integriteitsvereistes is noodsaaklik in die betroubare netwerk-interpretasie. Dié vereistes moet versekering gee dat inligting korrek vloei tussen komponente van die netwerk. Die vereistes sluit ook die volgende aspekte in soos byvoorbeeld:

(a) die korrektheid van boodskapversending,

(b) die waarmerking of sertifisering van die bron en eindpunt van die boodskap, en

(c) die korrektheid van verskeie datavelde wat gebruik word om oorplasing van gebruiker en protokol data te doen.

DIE ENKELBETROUBARE STELSEL-SIENING

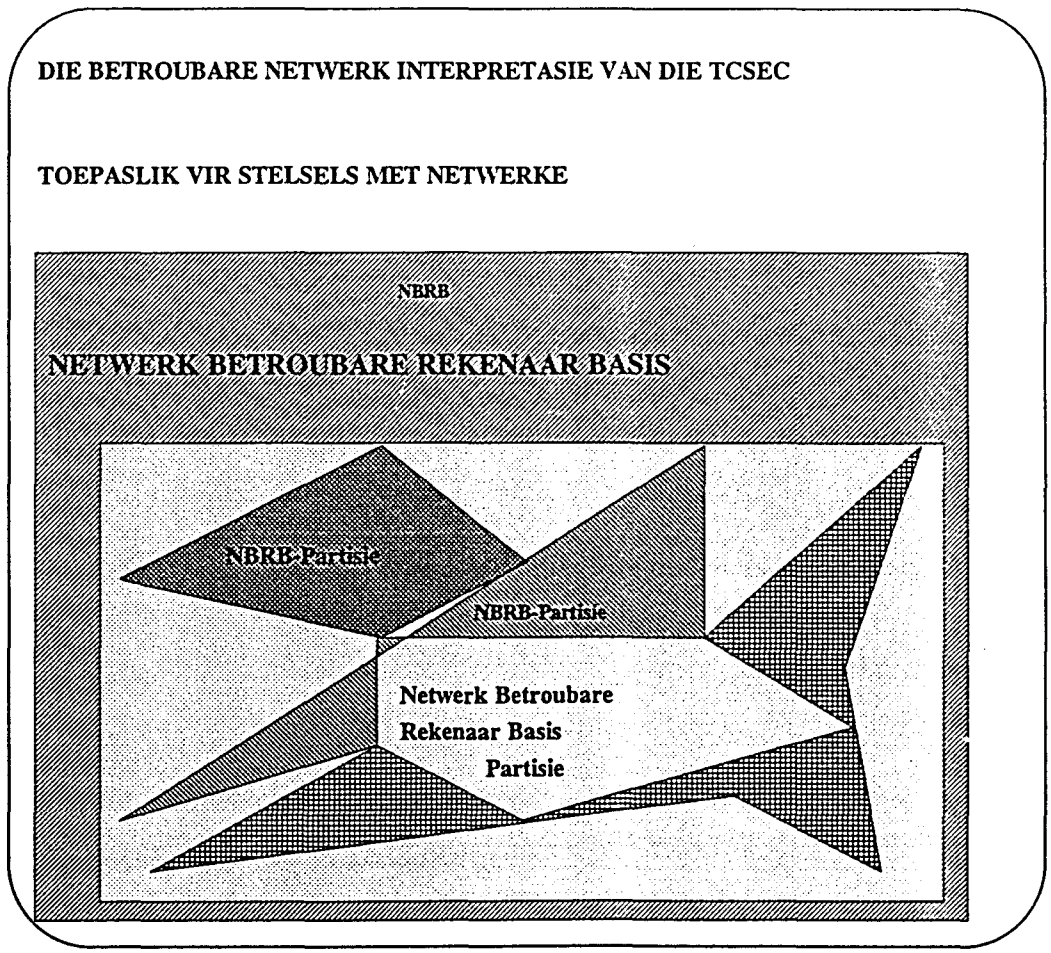


Fig 2.6. Die Enkel Betroubare Siening (Evalueringsmeganismes)

In die enkelbetroubare stelsel-siening (netwerk-evalueringsmeganismes), word die stelsel hanteer as 'n **volledige entiteit**.

Die Netwerkbetroubare Rekenaarbasis (NBRB) is die ekwivalent van die Betroubare Rekenaarbasis (BRB). Die NBRB mag in verskillende partisies of gedeeltes verdeel word, waar elk van hierdie partisies 'n Netwerkbetroubare Rekenaarbasispartisie genoem word. Elk van die partisies is verantwoordelik vir die handhawing van die sekerheidsbeleid vir die gedeelte van die stelsel waarin die partisie geleë is. Die Netwerkbetroubare Rekenaarbasis in geheel is verantwoordelik vir die handhawing van die algehele sekerheidsbeleid, d.w.s. dit wat vereis word vir die hele stelsel.

2.5.1. DIE NETWERKBETROUBARE REKENAARBASIS (NBRB)

Die versameling van NBRB partisies in geheel word as volledig beskou as dit gesamentlik die sekerheidsbeleid ondersteun. Die partisies kan egter elk verskillende dele van die sekerheidsbeleid ondersteun.

Die volledige NBRB mag afhanklik wees van die **protokolle** en **meganismes** wat gebruik word om betroubare en veilige kommunikasie tussen sy komponente te voorsien.

Individuele NBRB komponente ondersteun een of meer van die vier aspekte van 'n saamgestelde sekerheidsbeleid, naamlik :

(A) Verpligte Toegangsbeheer
(B) Diskresionêre Toegangsbeheer
(C) Identifikasie en Sertifisering
(D) Oudit

(A) VERPLIGTE TOEGANGSBEHEER

Die stelsel is voldoende as elke komponent in isolasie ondersteuning voorsien vir die volledige sekerheidsbeleid. Tans is die versekeringsvereistes vir stelsels wat 'n verpligte beleid ondersteun, slegs bereikbaar as die NBRB partisies disjunk is, d.w.s. alle ondersteuning van die verpligte beleid vir enige subjek moet plaaslik ondersteun word in die NBRB-gedeelte van die stelsel waarin die subjek val.

(B) DISKRESIONêRE TOEGANGSBEHEER

Diskresionêre Toegangsbeheermeganismes kan versprei word oor verskillende komponente van die stelsel, alhoewel dit die sekerheid dan sal beperk tot 'n C2 of 'n laer gradering[OSh91]. Die verskillende metodes wat gebruik kan word om diskresionêre sekerheid te voorsien is:

(i) GROEP-VAN-GEBRUIKER-beskerming :

Die beskermingsmeganisme ondersteun toegangsbeheer vir 'n aantal gebruikers, vanaf 'n afgeleë area.

Die aantal gebruikers bekend aan die bedienerstelsel word beperk.

In sulke gevalle word vereis dat die diskresionêre toegangsbeheer komponent van die kliëntstelsel individuele verantwoordelikheid vir gebruikers wat die netwerk dienste gebruik, voorsien.

(ii) AANGEE-beskerming :

Die aangee-beskermingstegniek omvat die aangee van gebruikersidentiteite tussen diskresionêre toegangsbeheermeganismes.

Die gebruikersidentiteit word gesertifiseer deur die kliëntmeganisme en daarna word dit aangegee aan die agent op so 'n wyse dat sy geheimhouding- en integriteitsversekering behoue bly.

(C) IDENTIFIKASIE EN SERTIFISERING

Identifikasie en sertifisering neem 'n nuwe voorkoms aan, want die sertifisering van eweknie-entiteite in die komponente van die Enkelbetroubare stelsel moet nou ook beskou word. Die sertifiseringsmeganismes moet eweknie-identiteite in die verskillende sekerheidstelsels korrek instel, so dat bedreigings soos die vervalsing van identiteit en die herspeel van die sertifiseringsprotokol nie oorgesien word nie.

'n Konneksie-georiënteerde model voorsien 'Eweknie-entiteitsertifisering' wat meestal voorkom as sertifiseringsprotokolle, wat beteken dat binding voorkom tussen die twee entiteite. Konneksielose metodes omvat data-oorsprongsertifisering waar die bron en die bestemming van kommunikasielyn benodig mag word om 'n per-boodskapbasis daar te stel. Geskikte tellermaatstawwe wat daargestel mag word in die ondersteunende protokolle sluit enkripsie skemas, tydstempels, handdruk, digitale handtekeninge en regverdigings skemas in.

D) INTEGRITEITSBELEID

'n Integriteitsbeleid adresseer beide pogings om inligting wat versend word en dan met opset verander word en pogings waar daar geen intensie is om veranderinge aan te bring nie, soos byvoorbeeld waar daar geraas in die kommunikasiestelsels voorkom of daar 'n toerustingfaling is. 'n Integriteitsbeleid lê hoofsaaklik klem op die vermoë om te skryf na objekte, asook om beperkte veranderinge aan inligting teweeg te bring.

Ondersteuning vir die integriteitsbeleid moet voorsien word deur die NBRB wat moet verseker dat die meganismes wat die integriteitsbeleid ondersteun, te alle tye beskerm word.

2.6. DIE INTERSKAKELENDE ERKENDE SIENING

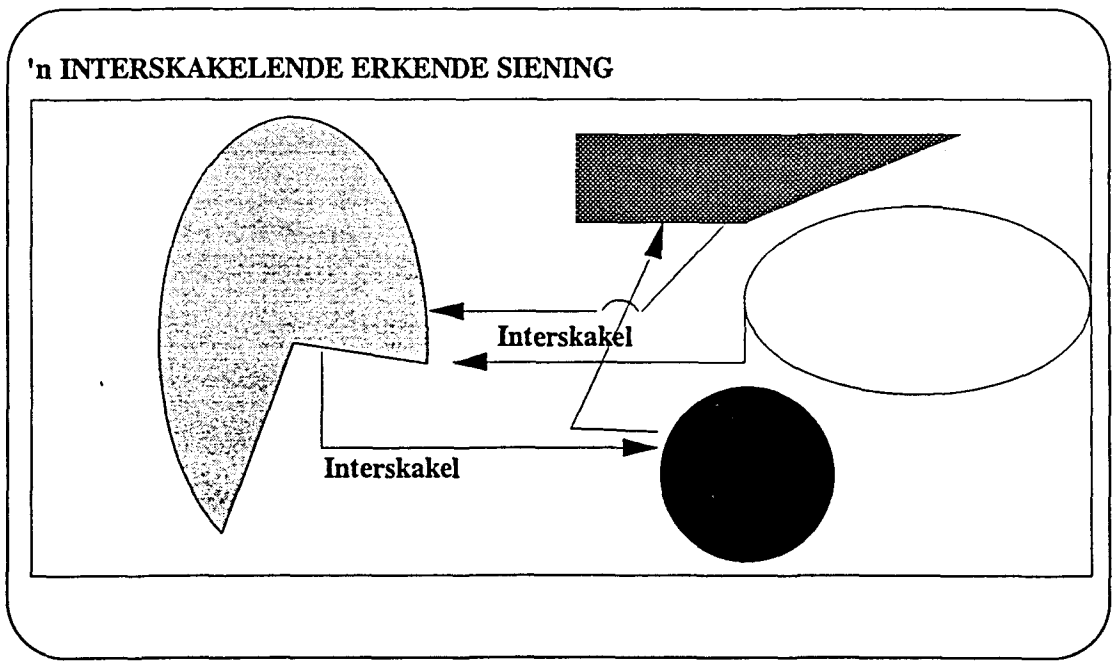


Fig 2.7 Die Interskakelende Siening(Netwerkevalueringsmeganismes)

Sekerheidstelsels word soms gesien as diskrete komponente wat losweg verbind word deur skakels of verbindings. Die sekerheidsgraadversekering is nie baie hoog in só 'n siening nie[OSh91]. Dié siening is van toepassing op stelsels wat nie met streng vereistes ontwerp en geïmplementeer word nie.

ITSEC

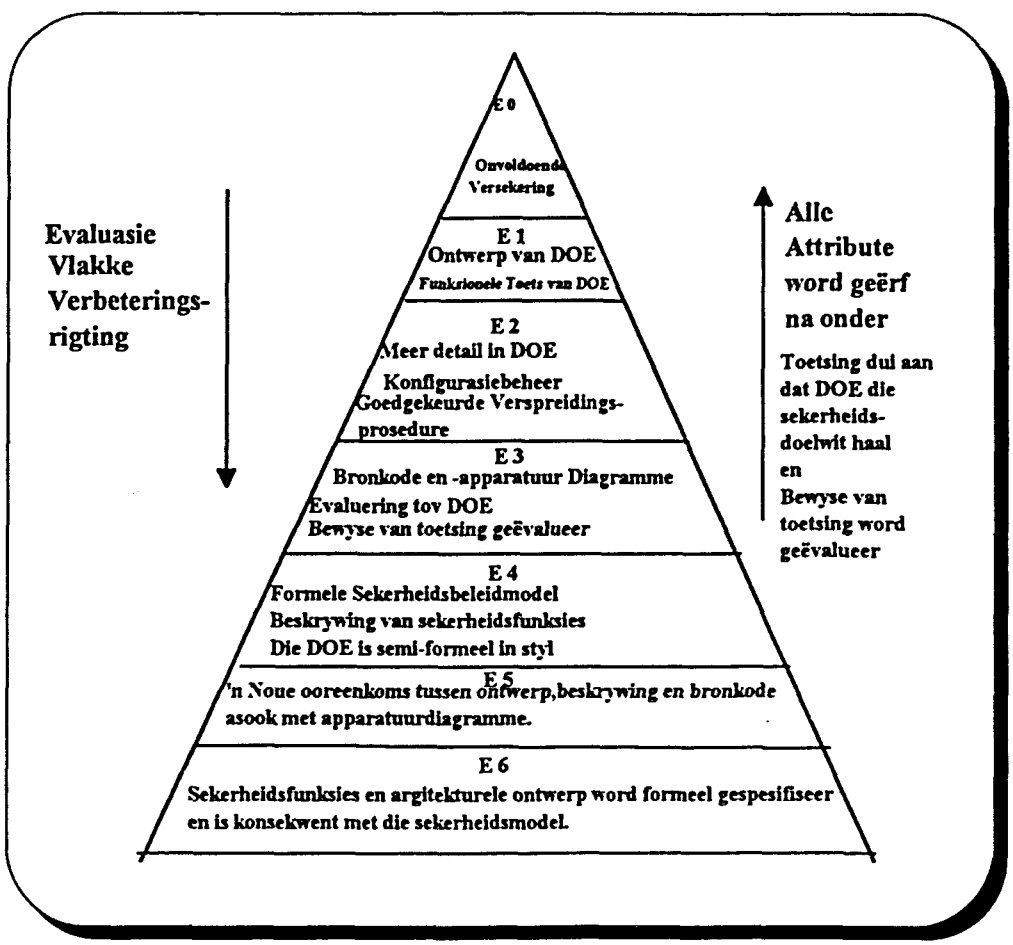
ITSEC skei die konsepte van **funksionaliteit** (Beskrywing van wat die stelsel doen) en van die **vertroue** in die korrektheid van die stelsel[OSh91].

Die doel van die ITSEC evalueringsproses is om 'n **STELLING VAN VERSEKERING** te voorsien. Die stelling van versekering omvat die vermoë van 'n produk of stelsel om sy **DOELWIT VAN EVALUASIE** (DOE - dieselfde konsep as die sekerheidsbeleid) te haal.

Daar word op twee gronde onderskei tussen sewe vlakke van versekering in die ITSEC-benadering. Die eerste is **korrektheid** wat grootliks 'n stelling is van hoe streng die stelsel ontwikkel en getoets is en die tweede aspek is dié van **effektiwiteit**, wat 'n stelling is van óf die sekerheidsfunksies geskik is en effektief saamwerk of nie, asook aspekte soos die hoeveelheid weerstand wat dit kan bied teen direkte aanvalle of versteekte gevalle, watter swakpunte daar bestaan, hoe maklik dit is om te gebruik en of die bekende swakhede gemanipuleer kan word in die praktyk.

Die ITSEC-korrektheidsaspekte het betrekking op die **deeglikheid en besonderhede** wat toegepas word op die ontwikkeling en werking van die DOE. Dit neem toe op elke vlak, insluitende alle kriteria van die laer vlakke.

Die vlakke lyk soos volg :



FIGUUR 2.8. DIE VLAKKE VAN DIE EVALUERINGSTEGNIEK ITSEC.

Let in fig 2.8. noukeurig op die onderskeie attribute van die verskillende vlakke vir meer besonderhede.

Die bogenoemde gedeelte het die verskillende evalueringsmeganismes wat gebruik kan word in die klassifisering van 'n sekerheidstelsel saamgevat. Deur gebruik te maak van hierdie evalueringsmeganismes as riglyne in die bou van 'n nuwe sekerheidsmodel kan 'n baie doeltreffender sekerheidsmodel saamgestel word. Nadat die model gebou is kan dit dan ook meer suksesvol geevalueer word, wat beteken 'n sekerheidstelsel met beter beheermeganismes kan voorsien word.

GEVOLGTREKKING

In die verkryging van rekenaarsekerheid in 'n omgewing is dit belangrik om eerstens te verstaan watter tipe bedreigings daar in die omgewing bestaan voordat daar 'n poging aangewend kan word om hierdie bedreigings hok te slaan. In die verkryging van rekenaarsekerheid in 'n omgewing kan daar volgens 'n doelgerigte plan te werk gegaan word. Dié plan word in sewe stappe uiteengesit en omvat onder andere 'n risiko-evaluering, die daarstelling van 'n sekerheidsplan en sekerheidsvereiste, die seleksie van meganismes wat gebruik gaan word in die implementering van die sekerheidsplan, die implementering van die sekerheidstelsel en die gereelde monitorbewegings. Die sekerheidsplan speel veral 'n belangrike rol in hierdie proses en daarom word daar heelwat aandag aan geskenk. 'n Sekerheidsplan op sigself bestaan uit 'n sekerheidsbeleid, die stel van die huidige toestand van die omgewing, aanbevelinge, 'n verantwoordelikheidsopdeling en 'n tydtabel. Die sekerheidsbeleid word gebruik in die sekerheidsmodel wat uiteindelik geïmplementeer word as die sekerheidstelsel, daarom is dit belangrik om die verskillende modelle wat tans bestaan, te verstaan voordat enige nuwe sekerheidsmodelle gebou kan word. Modelle wat onder andere in hierdie gedeelte bespreek word, is die verwysingsmonitor, die inligtingsvloeimodel, die Bell-Lapadula-model, die Biba-model, Graham-Denning-model en die Harrison-Ruzzo-Ullmanmodel.

Die evaluering van 'n sekerheidstelsel kan gedoen word deur gebruik te maak van verskeie meganismes, onder andere die TCSEC-evalueringsmeganisme, die ITSEC- en die NETWERK TCSEC-evalueringsmeganismes. Deur gebruik te maak van hierdie meganismes kan die nuwe sekerheidsmodel wat ontwikkel word, geëvalueer word. Deur die bestudering van die vereistes in die evalueringsmeganismes kan riglyne saamgestel word vir die bou van 'n betroubare en effektiewe sekerheidsmodel.

Die volgende stappe in die verkryging van 'n sekerheidstelsel is die uitsoek van meganismes wat gebruik kan word in die implementering van die sekerheidsmodel wat hanteer sal word in hoofstuk drie.

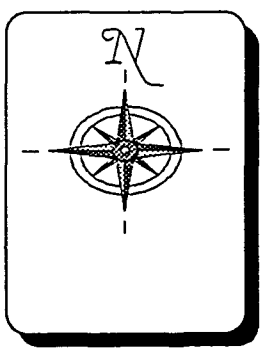
---oOo---

HOOFSTUK 3

REKENAARSEKERHEID - GRONDBEGINSELS EN MEGANISMES

3.1. INLEIDING

Die beskrywing van **Rekenaarsekerheid** in hoofstuk twee kan saamgevat word as die beskerming van die drie bates van 'n Rekenstelsel naamlik DATA, APPARATUUR en PROGRAMMATUUR. Gestel dat die sekerheidsplan op hierdie tydstip reeds uiteengesit is, en dat 'n omvattende sekerheidsbeleid en -model die sekerheidsplan ondersteun. Die sekerheidsmodel is dus nou gereed vir implementering en dus is die volgende stap in die verkryging van sekerheid die seleksie van sekerheidsmeganismes wat gebruik gaan word in die bou en implementeringsproses.



Hoofstuk drie beskryf die volgende tipes beskermingsmeganismes: Toegangsbeheermeganismes, geheuebeskermingsmeganismes en databasissekerheidsmeganismes. Toegangsmatrikse, gidlyste, toegangslyste, vermoëgebaseerde toegangsbeheer en proseduregeoriënteerde beheermeganismes word onder andere hanteer. Let veral in hierdie hoofstuk op die gebruik van die vermoëgebaseerde toegangsbeheer want dit speel 'n belangrike rol in DISMOD.

3.2.SEKERHEIDMEGANISMES

Beskermingsmeganismes kan verdeel word in algemene beskermingsmeganismes en spesifieke meganismes vir onder andere die volgende: Databasisse, Kommunikasie, Lêers, Geheue en Programme. Kommunikasie en verwerking vind gewoonlik plaas deur 'n netwerk met die bedryfstelsel en daarom lê die grootste gedeelte van die beskermingsmeganisme in die bedryfstelsel.

Meganismes in die bedryfstelsel sluit onder andere die volgende in: Toegangsbeheerlyste/-matrikse, Vermoëns, Proseduregeoriënteerde Toegangsbeheermeganismes, sowel as Verplasing, Basis-/Grensregisters, Etiketargitektuur, Segmentering en Paginering vir Geheuebeskerming. Gebruikers sertifisering word ook deur die bedryfstelsel hanteer. In die geval van databasisse word daar meer sekerheidsmeganismes benodig om die semantiese aspekte van die Databasis te beheer, bv. die volgende : Ouditmeganismes, Toegangsbesluitneming of Reëls, Enkripsie, Integriteitslotte, 'n Betroubare Voorverwerker, 'n Kommunikasiefilter, Sienings, ens.

Elk van die bogenoemde elemente sal kortliks bespreek word maar eerstens word 'n paar algemene terme gedefinieer om die gebruik daarvan in dié verhandeling duidelik te maak.

3.3. TERMINOLOGIE

OBJEKTE

Daar word oor die algemeen in sekerheidsdokumente en -boeke verwys na die beskerming van objekte, en daarom kan daar gevra word : ***WAT IS 'n OBJEK*** ?

'n Objek is 'n element wat of werklik bestaan, soos bv. 'n register of 'n skyfaandrywer, of dit kan abstrak wees, bv. 'n lêer of databasisrelasie. Tipes objekte verskil tussen die tipes omgewings wat beskerm moet word. In die meeste omgewings kan objekte geïdentifiseer word deur 'n veeltal, naamlik: 'n (Proses, Domein) veeltal waar die proses geïnisieer word deur 'n geïdentifiseerde gebruiker van die stelsel.

Daar sal vervolgens na OBJEKTE verwys word as ENTITEITE, om die verwarring wat kan voorkom met die gebruik van OBJEKTE in objekgeoriënteerde programmering te verhoed.

DOMEIN

'n Domein verteenwoordig 'n omgewing wat beskerm moet word. Die entiteit wat beskerm word, is eintlik die samestelling van die geïdentifiseerde subjek wat die domein gebruik en die domein, daarom word daar selde verwys na die domein.

REGTE

In die afdwing van sekerheid sal die gebruik van REGTE baie teëgekom word. Regte definieer die moontlike of toelaatbare aksies wat deur 'n subjek uitgevoer mag word, asook die moontlike regte wat met 'n entiteit geassosieer word[Pfl89].

AKTIEWE BESKERMINGSMEGANISMES

Aktiewe beskermingsmeganismes verhoed dat toegang tot 'n entiteit voorkom as die toegang nie as "gemagtig" aan die beskermingsmeganisme bekend is nie. 'n Voorbeeld hiervan is byvoorbeeld geheuebeskerming, waar toegang tot objekte op sekere adresse beheer kan word, afhangende van sekere kriteria. Die kriteria sluit byvoorbeeld die huidige verwerkertoestand in, asook attribute van 'n proses (vb.poging om die entiteit te gebruik)[Osh91] in.

PASSIEWE BESKERMINGSMEGANISMES

Passiewe beskermingsmeganismes is daardie meganismes wat ongemagtigde gebruik van inligting wat met 'n entiteit geassosieer word, verhoed of dit waarneem in gevalle waar toegang tot die objek nie verhoed word nie. Dié tegnieke word gebaseer op kriptografiese geheimhoudingstegnieke om ongemagtigde blootstelling van inligting te verhoed[OSh91].

3.4. BESKERMINGSMEGANISMES - TOEGANGSBEHEER TOT ENTITEITE

Die beskermingsmeganismes wat veral deur die bedryfstelsel geïmplementeer word, is, onder andere, gidslyste, toegangslyste, toegangsbeheermatrikse, vermoë- en proseduregeoriënteerde toegangsbeheermodules. Dié meganismes speel ook 'n belangrike rol in netwerke en databasisomgewings. In hierdie gedeelte moet veral gelet word op die sekerheidsmeganismes: vermoëns en proseduregeoriënteerde modules, omdat die meganismes 'n belangrike rol speel in DISMOD.

3.4.1. GIDSLYSTE

'n Gidslys is 'n eenvoudige beskermingsmeganisme. Veronderstel die Domein wat beskerm word deur gidslyste bestaan uit 'n versameling subjekte S en 'n versameling entiteite E.

'n Gidslys is 'n lys waarin alle toegange gelys word wat 'n subjek tot verskillende entiteite besit. Elke subjek in 'n domein besit 'n gidslys. Let op die voorstelling van gidslyste soos gesien in figuur 3.1. vir verdere verduideliking.

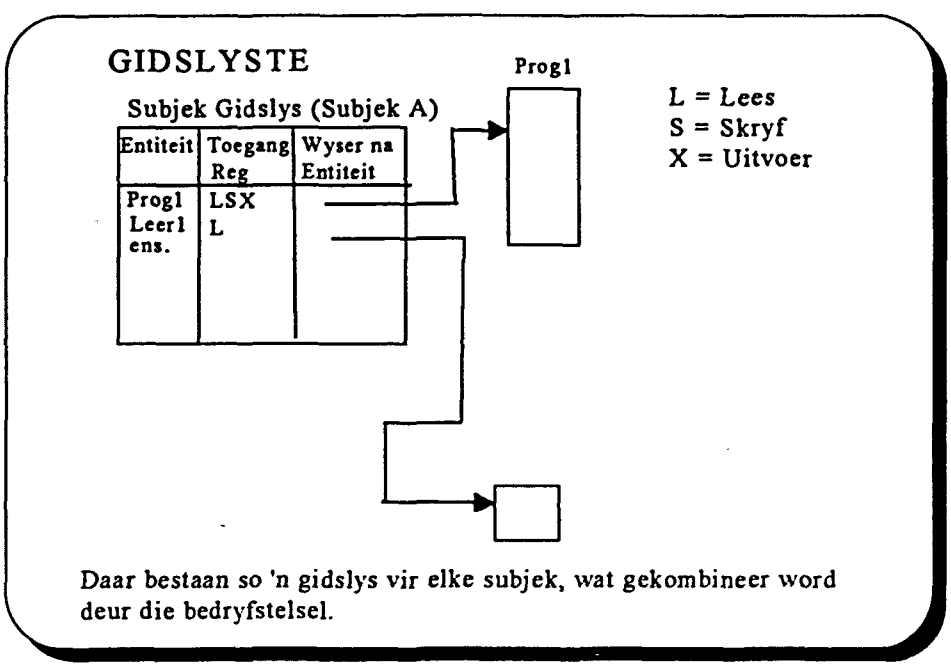


Fig 3.1. Gidslyste.

Enige subjek S wat 'n entiteit skep in die domein word die Eienaar van die entiteit genoem. Die eienaar van 'n entiteit mag besluit watter tipe toegang ander subjekte tot sy entiteit mag besit. Dit wil sê indien die eienaar van 'n entiteit (sê entiteit A), toegangsreg aan 'n ander subjek(sê subjek B) wil gee tot sy entiteit, sal hy dié versoek aan die bedryfstelsel rig, wat weer op sy beurt 'n inskrywing sal maak in subjek B se gidslys, wat sal aandui dat subjek B nou die spesifieke toegang tot entiteit A besit.

Alle gidslyste word in die bedryfstelsel gehou vanwaar enige versoek vir toegang tot 'n entiteit geverifieer word.

Alhoewel gidslyste die eenvoudigste sekerheidsmeganisme is, het dit ongelukkig sekere nadele, naamlik :

(a) Die aantal inskrywings in 'n gidslys word te veel indien daar 'n inskrywing in die gidslys gemaak word vir elke publieke, of gedeelde entiteit.

(b) Die wegneem van toegangsregte van byvoorbeeld publieke entiteite word 'n probleem want elke gidslys wat bestaan, moet nou deursoek word vir 'n inskrywing van die betrokke entiteit, voordat dit geskrap kan word. Dit beïnvloed die kwaliteit van die sekerheidsmeganisme,

(c) Die laaste probleem kom voor met entiteitidentifiseerders. Die feit bestaan dat verskillende subjekte, entiteite in hul besit met dieselfde identifiseerders kan identifiseer. Subjek A mag byvoorbeeld 'n entiteit met die naam A besit, maar subjek B kan 'n ander entiteit besit wat ook A genoem word. Indien beide subjek A en B nou toegangsreg aan subjek C vir entiteit A gee, moet die bedryfstelsel kan identifiseer watter entiteit A in subjek C se gidslys, subjek C wil gebruik.

3.4.2. TOEGANGSBEHEERLYS

'n Gidslys behoort aan 'n subjek en is die lys van toegange wat die betrokke subjek tot spesifieke entiteite besit, waar 'n toegangsbeheerlys net die teenoorgestelde is. 'n Toegangsbeheerlys behoort aan 'n entiteit en is 'n lys van toegange wat besit word deur subjekte vir 'n betrokke entiteit.

'n Toegangsbeheerlys lyk dus soos volg (fig 3.2):

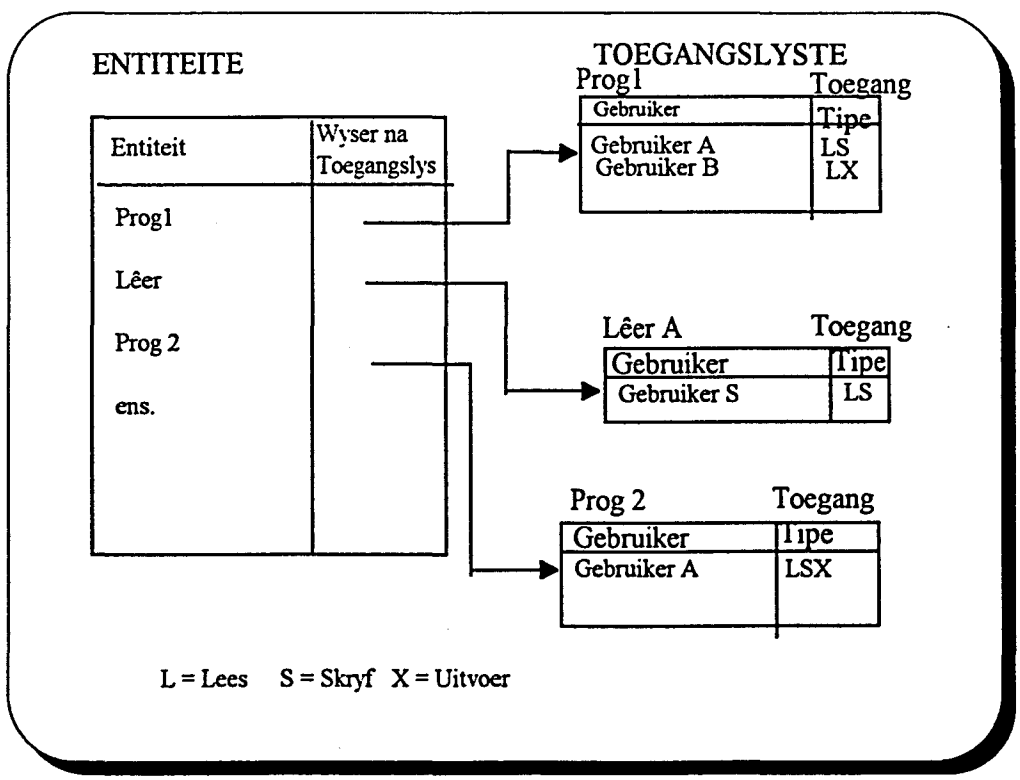


Fig 3.2. Die Toegangslys.

Toegangsbeheerlyste, soos gesien in figuur 3.2. mag ook verstekinskrywings besit vir enige subjekte wat beteken dat sommige gebruikers spesifieke toegang tot die entiteit kan besit terwyl al die ander subjekte wat nie spesifiek genoem word nie, die verstektoegang tot die entiteit besit. 'n Voorbeeld van 'n bedryfstelsel wat 'n toegangbeheerlys implementeer is MULTICS.

MULTICS maak ook gebruik van beskermingsklasse vir die definiëring van toegangsregte. In MULTICS word drie beskermingsklasse gebruik, naamlik gebruiker, groep en kompartement. Indien 'n subjek aanteken, teken hy as 'n kombinasie van die drie beskermingsklasse aan, en sal toegangsregte kry volgens hierdie aantekenkombinasie. Die inskrywings in die toegangsbeheerlys is in die vorm van 'n toegangreg vir 'n spesifieke drietal <gebruiker,groep,kompartement>.

In die gidslyste word ook gebruik gemaak van "wildcards", byvoorbeeld in die aanteken drietal <gebruiker A,*,kompartement A>, sal die subjek wat op hierdie wyse aangeteken het, toegangsreg verkry tot alle entiteit waar toegang gegee word aan gebruiker A spesifiek, maar enige groep en kompartement A spesifiek.

Die voordeel van die gebruik van toegangsbeheerlyste is dat "Naambeskermde Binding"[OSh91] plaasvind. Naambeskermde binding is binding wat plaasvind op die laaste moontlike oomblik, d.w.s. op die tydstip as die versoek gerig word. Naambeskermde binding verbeter die beheer van verandering aan die magtigingstoestand[OSh91], wat beteken dat as toegangsregte verander het tussen twee toegangsversoeke die veranderde magtigingsvereiste onmiddellik geïmplementeer word.

3.4.3. TOEGANGSBEHEERMATRIKSE

'n Toegangsbeheermatriks is in der waarheid die kombinering van gidslyste en toegangsbeheerlyste. Dit is 'n twee-dimensionele tabel, waar die horisontale rye subjekte identifiseer, en vertikale rye of kolomme die entiteite identifiseer. Die selle of afbeeldings van rye en kolomme (subjek,entiteitnaam) hou rekord van die regte of toegangsregte van die subjek tot die entiteit in aanvraag. Die toegangsbeheermatriks lyk soos volg:

Objekte / Subjekte	Lêer A	Prog 1	Prog 2	....
Gebruiker A	LS		X	
Gebruiker B		LSX		
Gebruiker C	L	LX	LSX	
...				

L= Lees , X = Uitvoer, S = Skryf
Fig 3.3. Die toegangsbeheermatriks

Die onderstaande bewerking is 'n versameling van bewerkings wat uitgevoer mag word om die matriks te verander. Die fundamentele bewerkings sluit gewoonlik die volgende in:

Die BYVOEGING van 'n nuwe subjek
Die SKRAPPING van 'n subjek
Die BYVOEGING van 'n nuwe entiteit
Die SKRAPPING van 'n entiteit
Die BYVOEGING van 'n Toegangsreg
Die VERWYDERING van 'n Toegangsreg

Toegangsbeheermatrikse identifiseer entiteite wat gebruik kan word deur 'n enkele subjek waar toegangslyste die subjekte identifiseer wat 'n enkele entiteit kan gebruik[Pfl89]. Die soektog deur 'n groot aantal van die drietalle [subjek,entiteit,reg] wat 'n inskrywing in die matriks verteenwoordig, is oneffektief en word selde gebruik[Pfl89].

3.4.3.1. EKSKLUSIEF-UITSLUITENDE TOEGANGSBEHEERMATRIKS

'n Eksklusief-uitsluitende toegangsbeheermatriks lyk presies dieselfde as 'n toegangsbeheermatriks. In die gewone toegangsbeheermatriks dui 'n inskrywing op toegang tot 'n objek, waar die inskrywing in 'n eksklusief

uitsluitende toegangsbeheermatriks dui op weiering van toegang. Dit word byvoorbeeld gebruik in die SEAVIEW model[Lun89].

3.4.4. VERMOëGEBASEERDE MEGANISMES

'n VERMOë is 'n onvervalsbare teken of sleutel wat aan die eienaar sekere regte tot 'n entiteit gee[Pfl89]. O'Shea, et al[Osh91] vergelyk die vermoë met 'n ry van die toegangsmatriks, wat al die regte van 'n subjek op alle entiteite, waartoe die subjek toegang besit, aandui. 'n Vermoë is ook 'n entiteit wat beskerming vereis, daarom word dit óf in 'n geëtiketteerde bergingsarea gestoor, of in 'n onadresseerbare deel van die geheue. Die ander wyse wat daar bestaan om 'n vermoë te beskerm, is deur dit te beskerm met 'n ander vermoë. (DISMOD)

'n Vermoë kan vergelyk word met 'n identiteitsdokument of 'n teaterkaartjie, ens., omdat beide hierdie elemente getoon moet word om toegang te verkry. Die feit dat 'n vermoë toegang waarborg tot 'n entiteit maak dit dus soveel belangriker dat die **vermoë nie gedupliseer moet kan word nie.** 'n Vermoë is dus 'n kaartjie wat toestemming gee aan 'n subjek om 'n sekere tipe toegang tot 'n entiteit te verkry.

Een manier om 'n vermoë onvervalsbaar te maak, is om die vermoë nie direk aan die gebruiker te gee nie maar om eerder alle vermoëns in die bedryfstelsel namens die subjekte te hou. Dit kan verseker word deur te spesifiseer dat 'n vermoë slegs geskep kan word deur 'n spesifieke navraag aan die bedryfstelsel. Elke vermoë identifiseer toelaatbare suksesse.

'n Subjek wat nuwe entiteite skep, moet die bewerkings wat toelaatbaar is op hierdie entiteit spesifiseer, d.w.s. nie net lees, skryf, bywerk of skrap nie, maar ook nuwe tipes toegange kan gespesifiseer word. Dié toegangstipes word dan gebruik om 'n vermoë vir die entiteit te skep.

Berging van vermoëns

Vermoëns kan onder andere beskerm word deur gebruik te maak van etikettering. Etikettering is 'n tegniek waar bergingsruimtes wat vermoëns allokeer, geëtiketteer word met 'n onadresseerbare veld wat aandui dat die area 'n vermoë bevat. Die ander manier waarop vermoëns beskerm word, is deur dit te berg in beskermde areas van die geheue. Die areas word "vermoë-segmente"[OSh91] genoem. Geheuebeskermingsmeganismes, byvoorbeeld basis-grens-segmentering, word gebruik om die vermoë-segmente te kontroleer en hul integriteit te waarborg[OSh91,Pfl91]. In die tweede vorm van beskerming word vereis dat vermoëns geskep en geïnterpreteer word deur programmatuur[OSh91].

Werking van vermoëns

'n Voorbeeld van 'n tipe toegangsreg wat verkry kan word vir 'n entiteit is oorplaasbaarheid of verspreibaarheid van die entiteit. 'n Subjek wat hierdie tipe toegang (oorplaas, versprei) tot 'n entiteit besit, mag kopieë van vermoëns aan ander subjekte toeken of aangee in plaas daarvan om 'n kopie van die entiteit aan die ander subjek te gee. Elk van hierdie tipes vermoëns wat toegeken of aangegee word, besit 'n lys van toelaatbare toegangstipes vir die entiteit, waarvan een die oorplaastipereg is.

'n Voorbeeld van die gebruik van 'n vermoë waar die kopiëring van 'n entiteit vereis word, is die volgende:

Proses A kan byvoorbeeld 'n kopie van die vermoë vir entiteit O aangee vir subjek B, wat weer dié vermoë na subjek C kan aangee. In die aangee van die vermoë, bly alle toegangstipes dieselfde in die vermoë, d.w.s. die toegangstipe <mag-vermoë-oorplaas> in die vermoë word ook oorgeplaas. B kan verhoed dat die vermoë verder versprei deur die oorplaas reg weg te laat uit die regte wat aangegee word in die vermoë na C toe. B mag in hierdie geval steeds slegs sekere toegangsregte na C aangee, maar nie die reg om die vermoë na ander subjekte aan te gee of te versprei nie.

As 'n proses uitgevoer word, werk dit in 'n domein of plaaslike naamruimte. 'n Domein is die versameling van entiteite waartoe die proses wat uitgevoer word, toegang besit. 'n Domein vir 'n gebruiker op 'n gegewe tyd mag programme, lêers, datasegmente en I/O-toestelle insluit.

Met die uitvoering van 'n prosedure mag dit gebeur dat die prosedure 'n subprosedure roep, en sommige van die entiteite waartoe die subjek toegangsreg besit, kan dan as argumente aangegee word na die subprosedure. Die domein van die subprosedure is nie noodwendig dieselfde as die van die roepende prosedure nie, inteendeel die roepende prosedure mag slegs sommige van sy entiteite na die subprosedure aangee, en die subprosedure mag dalk weer toegangsregte besit tot entiteite wat nie toeganklik is vir die roepende prosedure nie. Die roeper mag dalk ook net sommige van sy toegangsregte of vermoëns vir die entiteite wat dit aangee, na die subprosedure aangee. Die bedryfstelsel skep dan nuwe vermoëns vir die subprosedure wat dit in staat sal stel om die proses te voltooi indien die subprosedure die nodige regte besit.

Vermoëns is eenvoudig en dit is maklik om rekord te hou van die toegangsregte van subjekte tot entiteite gedurende uitvoertyd. Vermoëns kan gerugsteun word deur 'n meer omvattende tabel, soos 'n toegangsbeheermatriks of 'n toegangsbeheerlys. Elke keer as 'n proses 'n nuwe entiteit wil gebruik, ondersoek die bedryfstelsel die meesterlys van entiteite en subjekte om te bepaal of die entiteit toeganklik is. As dit so is, skep die bedryfstelsel 'n vermoë vir die entiteit.

Gedurende uitvoering word slegs die vermoëns van entiteite wat gebruik is gedurende die huidige proses beskikbaar gehou. Dié beperking verbeter die spoed waarmee toegang na 'n entiteite getoets word.

VOORDELE VAN VERMOëNS

Belangrike voordele van vermoëmeganismes is hul betroubaarheid en effektiwiteit. Die voorstelling en interpretering van 'n vermoë is effektief, want die vermoë kan

maklik gevind word as dit nie onmiddellik beskikbaar is nie. Tweedens is dit nie nodig om naamevaluering van die subjek uit te voer met elke toegangspoging nie, maar slegs as die vermoë gegenereer word. Indien 'n vermoë direk verwys na 'n entiteit eerder as na 'n benoemings-stelsel, dan word naamevaluering geëlimineer met uitvoertyd wat meer effektiwiteit voorsien.

PROBLEME

Wegneem van regte

'n Probleem is dat vermoë weggeneem kan word. Die probleem wat ontstaan, is dat daar vermoëns bestaan wat nog steeds uitdeelregte vir die weggeneemde vermoë tot 'n entiteit besit. Die oplossing tot dié probleem is om

(a) 'n Tru-wyser na vermoëns te hê,

(b) 'n Periodiese skrapping en heruitdeling van vermoëns te doen

(c) 'n Gesentraliseerde vermoëlys te skep

Die laasgenoemde tegniek, is 'n tegniek wat toelaat dat aspekte soos wegneming, magtigingstoestandverandering, ens. vereenvoudig word.

'n Gesentraliseerde vermoëlys sal vervolgens uiteengesit word, volgens fig 3.4

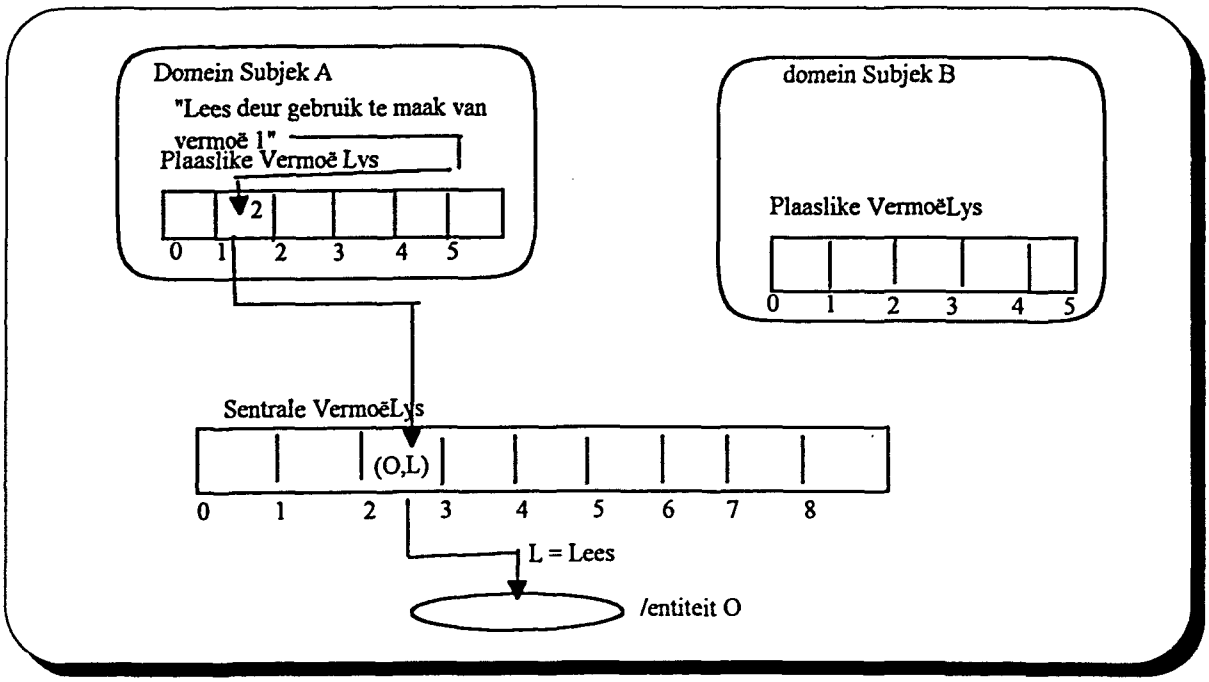


Fig 3.4. Die Sentrale VermoëLys

Elke subjek ontvang nou 'n vermoëlys wat net plaaslik is tot daardie subjek, maar elke vermoë in dié subjek se vermoëlys wys weer na 'n vermoë in die sentrale vermoëlys, wat in der waarheid die eintlike toegang na die objek sal voorsien. In die geval van 'n sentrale vermoëlys is daar nou net een vermoë wat weggeneem sal moet word, naamlik die vermoë in die sentrale vermoëlys. Daar sal tog dan in hierdie geval gereelde skoonmaakprosesse moet plaasvind waar alle wysers in die plaaslike vermoëlyste skoongemaak word indien daar nie meer 'n vermoë daarvoor in die sentrale vermoëlys bestaan nie.

3.4.5. PROSEDURE-GEORIëNTEERDE TOEGANGSBEHEER

'n Ander meganisme wat gebruik word vir toegangsbeheer is die proseduregeoriënteerde toegangsbeheer metode.

Die doelwit in dié tipe toegangsbeheer is - om nie net beperkings te plaas op dit waartoe 'n subjek in 'n entiteit toegang besit nie, maar ook tot dit wat die subjek aan die entiteit kan doen. Lees- teenoor skryftoegang na 'n entiteit kan nogal maklik beheer word, maar meer komplekse beheer is nie maklik om te verkry nie.

Prosedure-georiënteerde beskerming is die beskermingsmetode waar daar 'n prosedure bestaan wat toegang tot die entiteit beheer. Die prosedure vorm 'n kapsule rondom die entiteit, en laat so gespesifiseerde toegang tot die entiteit toe. 'n Ander eienskap van proseduregeoriënteerde beskerming is dat die prosedure kan vereis dat toegang tot 'n entiteit gedoen word deur 'n betroubare koppelvlak.

Proseduregeoriënteerde beskerming implementeer die beginsel van inligting-versteking, omdat die wyse van implementering na 'n entiteit bekend staan slegs aan die beheerprosedure van die entiteit. Die graad van beskerming dra 'n nadeel van effektiwiteit. Daar kan geen eenvoudige vinnige toegang tot 'n entiteit bestaan selfs as die entiteit gereeld gebruik word nie[Pfl89].

3.4.6. SLOT- EN SLEUTELMEGANISMES

Die laaste meganismes vir toegangsbeheer is die slot- en sleutelmeganismes wat soos volg werk:

'n "Entiteit-identifiseerder-sleutel" paar word geassosieer met 'n subjek en

'n Lys van "Sleutel-Toegangsreg"-pare word geassosieer met elke entiteit wat beskerm word.

Indien 'n subjek toegang wil verkry tot 'n beskermde entiteit, toon die subjek net sy "Entiteit-identifiseerder-sleutel"-paar aan die beskermingsmeganisme, wat weer die gegewe sleutel en toegangsbeheermodus vergelyk met die inskrywings in die entiteit se "sleutel-Toegansreg" pare om 'n passing te verkry. Indien daar nie 'n passing bestaan nie word toegang geweier andersins word dit toegelaat[OSh91].

'n Voorbeeld van die gebruik van hierdie meganisme is in die IBM-Stelsel 370 argitektuur[OSh91].

3.4.7. KRIPTOGRAFIE

Kriptografiese stelsels beskerm inligting deur middel van data transformasies. Die transformasies word enkriptering genoem as die transformasie vanaf die oorspronklike skoonteks na die syferteks omgeskakel word[G'Oshea] en dekripsie as die transformasie vanaf syferteks na die oorspronklike skoonteks gaan.

Kriptografie word ook gedefinieer as die studie van wiskundige skemas vir die oplossing van sekerheidsprobleme soos byvoorbeeld die vertroulikheid van data, waarmerking van boodskappe en waarmerking van individue[Lau92].
Kriptografiese skemas kan in vyf kategorieë verdeel word[Lau92], naamlik :

(a) Simmetriese enkripsieskemas

Simmetriese enkripsieskemas gebruik dieselfde enkripteer- en dekripteersleutel vir die sender en die ontvanger, d.w.s. die sleutel moet versprei word na beide partye op die veiligste moontlike manier. Die verspreiding van die sleutel is 'n nadeel van die meganisme, maar die voordeel van die meganisme is dat dit baie vinnig is.

(b) Asimmetriese enkripsieskemas

In assimetriese enkripsieskemas bestaan daar een sleutel vir enkriptering en 'n ander vir dekriptering. Die enkripteersleutel word publiek gemaak sodat enigeen wat 'n boodskap stuur aan die ontvanger dit kan enkripteer met die enkripteer sleutel. Die dekripteersleutel word egter geheim gehou by die ontvanger, wat dan die boodskap kan dekripteer. Die ontsyfering van die boodskap of dekripteer sleutel sal in hierdie geval te lank neem om effektief te wees vir die ontsyferaar en sleutels hoef nie versprei te word nie.

(c) Asimetriese handtekeningskemas

'n Digitale handtekening is 'n aantal bisse waarvan die waarde afhanklik is van die boodskap waaraan dit geheg word. 'n Handtekening vanaf 'n subjek A kan net deur A vervaardig word, d.w.s. die boodskap kan nie vervals word nie en indien die boodskap verander word, sal die handtekening nie meer ooreenstem met die boodskap nie. Die RSA-algoritme is 'n voorbeeld van 'n enkripsieskema wat 'n digitale handtekening kan genereer[Lau92].

(d) Interaktiewe bewyse

Interaktiewe bewyse word gebruik vir die waarmerking van subjekte. 'n Subjek S kan byvoorbeeld 'n geheime sleutel bevat wat verwant is aan 'n ander P. 'n Ander subjek, subjek B wil dan subjek S se identiteit bevraagteken. Subjek B sal nou aan subjek S 'n paar vra, vra wat subjek S met behulp van die geheime sleutel

stuur. Subjek B gebruik dan vir P om die antwoorde te verifieer. Indien alle antwoorde korrek is, word subjek S aanvaar as die korrekte subjek S.

(e) Assimetriese sleutelverspreidingskemas

Assimetriese sleutelverspreidingskemas word byvoorbeeld gebruik oor onveilige kommunikasielyne, deur data uit te ruil oor die lyn sodanig dat die sender en ontvanger 'n algemene geheime sleutel kan konstrueer uit die data.

3.5. DATABASISSEKERHEIDSMEGANISMES

Die Databasisbeheerstelsel voorsien 'n baie fyner beheermeganisme as die bedryfstelsel, omdat dit ook die semantiese waarde van die data in ag moet neem.

Die volgende onderafdelings behandel die grondbeginsels van databasissekerheid, nl. aspekte soos gesentraliseerde teenoor gedesentraliseerde beheer, eienaarskap teenoor administrasie, toegangsbeheerspesifikasies, ens.

3.5.1. GESENTRALISEERDE BEHEER TEENOOR GEDESENTRALISEERDE BEHEER.

Gesentraliseerde beheer beteken dat daar slegs een enkele magtiger (of groep) is wat alle sekerheidsaspekte beheer (INGRES) [Fer81]. In 'n Gedesentraliseerde beheermeganisme beheer verskillende administreerders verskillende gedeeltes van die databasis. Hierdie administreerders volg normaalweg riglyne wat neergelê is vir die hele organisasie.

3.5.2. EIENAARSKAP TEENOOR ADMINISTRASIE

Die eienaar van 'n databasis word beskou as die persoon wat verantwoordelik is vir die skepping van die data. 'n Voorbeeld kan bv. die volgende wees : as die salarisdepartement die enigste is wat die databasis bywerk, dan word die hoof van die salarisdepartement as die eienaar van die databasis beskou. Dit is baie keer moeilik om die eienaar te identifiseer. As die konsep van eienaarskap nie gebruik word nie, word die administrasiefunksie gebruik. Die doelwit van die administrasiefunksie is om die gedeelde data te definieer deur die gebruikers en om dan sy gebruik te beheer. Hierdie funksie kan verrig word deur die eienaar as daar een bestaan of deur die databasisadministrateur. Die verskil tussen hierdie twee beleidsrigtings lê in die feit dat die eienaar tot enige tipe toegang toegelaat word terwyl die administreerder slegs die regte besit om beheer oor die data uit te voer[Fer81].

3.5.3. BELEID VIR TOEGANGSBEHEERSPESIFIKASIE DIE NODIG-OM-TE-WEET-BELEID.

Dié beleid beperk inligting tot die gebruikers wat regtig die inligting benodig vir hul werk. Die beleid beskerm stelsels teen die uitlek van inligting en dit verminder die moontlikheid dat die integriteit van die databasis prysgegee sal word. Die beleid word ook soms die **beleid van die minste voorregte** genoem[Fer81].

MAKSIMEERDELING.

Die intensie van hierdie beleid is om maksimum gebruik van inligting te verskaf.

OOP- EN GESLOTE STELSELS.

Toegang word slegs toegelaat in 'n geslote stelsel indien daar eksplisiete magtiging bestaan.(Nodig-om-te-weet) In 'n Oopstelsel word toegang net geweier indien dit eksplisiet verbied word[Fer81].

NAAM-AFHANKLIKE TOEGANGSBEHEER

In Sekerheid moet ons ten minste in staat wees om die dataobjekte waartoe 'n gebruiker toegang het te spesifiseer. Die diepte van die dataobjekte in die toegangsreëls is 'n ander beleidsbesluit. 'n Streng interpretasie van die beleid van minste voorregte vereis dat die objekte die fynste beskerming toelaat deur die DBBS. Hierdie tipe beheer word **naam-afhanklike toegangsbeheer** genoem[Fer81].

KONTEKS-AFHANKLIKE BEHEER.

Die beleid van **konteks-afhanklike toegangsbeheer** verwys na kombinasies items. Een faset van die beleid beperk die velde waartoe gelyktydig toegang verkry kan word. 'n Ander aspek van die beleid is die vereiste dat sekere velde saam voorkom[Fer81].

3.5.4. BELEID OM INLIGTINGSVLOEI TE BEHEER.

Programgebruik van inligting of data moet beheer word teen die uitlek van inligting. Data kan bv. vloei van 'n gemagtigde program na 'n ongemagtigde program - dit moet verhoed word.

Daar word implisiet aangeneem dat 'n magtiger bestaan wat toegangsregte aan gebruikers gee. Dit staan bekend as ***diskresionêre toegangsbeheer***. 'n Eenvoudiger benadering sal wees om die data in kompartemente of kategorieë te plaas. In hierdie geval sal gebruikers van een kategorie nie toegang besit tot data in 'n ander kategorie nie. Dit is 'n voorbeeld van ***nie-diskresionêre toegangsbeheer***. 'n Uitbreiding van die kompartement beleid is die multivlak beheerbeleid (gewoonlik gebruik in militêre instansies). Hierdie beleid besit kategorieë en inligting word geklassifiseer in verskillende vlakke,bv. ONGEKLASSIFISEERD, KONFIDENSIEEL, GEHEIM en HOOGS

GEHEIM. 'n Sekerheidsvlak word dan gedefinieer as 'n klassifikasie en 'n versameling van kategorieë[Fer81].

Die lys van entiteite waartoe 'n subjek toegang besit, saam met die toegangstipe word soms die vermoëlys van die subjek genoem [Fer81].

3.6. BESKERMINGSMEGANISMES - GEHEUEBESKERMING EN -ADRESSERING

Geheuebeskermingsmeganismes speel 'n belangrike rol in multiprogrammering deurdat die tegnieke wat gebruik word, verhoed dat een program 'n volgende se geheueruimte affekteer. Let in hierdie gedeelte veral op die gebruik van die basis/grensbeskermingsmeganisme, want vermoëns word byvoorbeeld geïmplementeer met behulp van hierdie meganisme.

Die geheuebeskermingsmeganismes wat vervolgens bespreek gaan word, sluit onder andere die volgende in:

(a) Heinings
(b) Heralokering
(c) Basis/Grensregisters
(d) Geëtiketteerde Argitektuur
(e) Segmentering
(f) Paginering
(g) Gekombineerde gebruik van Paginering met Segmentering.

3.6.1. Heining-/Grenstegniek.

Die heining- of grenstegniek, soos die naam aandui, stel 'n adres in wat dien as heining of grens tussen die gedeelte van die geheue wat gebruik word vir die bedryfstelsel en die gedeelte van die geheue wat gebruik word vir gebruikers.

Figuur 3.1 is 'n voorbeeld van die grens tegniek. Dié grens word geïmplementeer op twee wyses:

(a) 'n Vaste grensadres, wat of geen ruimte vir die bedryfstelsel bied om te groei nie, óf geheueruimte mors, omdat die bedryfstelsel dit nie gebruik nie.

(b) 'n Register wat die grensadres bevat.
Beide die bogenoemde tegnieke beskerm die bedryfstelsel teen gebruikers, maar gebruikers word nie teen mekaar beskerm nie.

3.6.2. Heralokering

Heralokering is die proses waar 'n program só geskryf is dat dit voorkom asof dit begin by adres nul, en dan die verandering van alle adresse om die werklike adres waarop die program geallokeer word in die geheue te weerspieël. Dit beteken in die meeste gevalle dat 'n konstante heralokeringsfaktor net by elke adres in die program getel word. Dié heralokeringsfaktor is in die meeste gevalle die beginadres van die geheue wat toegeken word vir die program.

3.6.3. Basis-/Grensregisters.

Meervoudige gebruikers in 'n stelsel vereis dat elke gebruiker se programme geallokeer sal word in sy eie geheueruimte. 'n Basisregister word gebruik as heralokeringsregister vir elke gebruiker se program, d.w.s. die addresse in elke gebruiker se program is die basisregister-adres + die adres in die gebruiker se program. 'n Tweede register, die grensregister, dien as boonste adreslimiet vir 'n program. Die adresse van 'n gebruikersprogram mag dus nie buite hierdie limiete val nie.

'n Gebruiker word beskerm van buitegebruikers oftewel foute in ander gebruikers se programme met die instelling van basis-/grensregisters. Al instruksies wat die gebruiker kan beïnvloed, is instruksies wat die instruksieadresruimte van die

program wil oorskryf. 'n Oplossing hiervoor is om basis-/grensregisters vir beide die instruksies en die data van die program te verskaf.

3.6.4. Geëtiketteerde Argitektuur

In 'n geëtiketteerde argitektuur het elke woord- of masjiengeheue een of meer ekstra bisse wat die toegangsregte tot daardie woord identifiseer. Dié toegangsbisse word gestel deur voorreg-instruksies(bedryfstelsel). Die bisse word getoets elke keer as daardie adresruimte gebruik word. Die tipes beskerming wat gestel kan word in die toegangsbisse is byvoorbeeld Lees,Skryf, Uitvoer-alleenlik, ens.

'n Variasie van die bogenoemde wat ook gebruik word, is waar 'n groep van opeenvolgende adresruimtes geëtiketteer word met een etiket vir die groep adresse.

3.6.5. Segmentering

Segmentering is wanneer 'n program opgedeel word in aparte gedeeltes, waar elke gedeelte nie net 'n logiese eenheid besit tussen die gedeeltes nie, maar ook 'n verwantskap tussen al die kode of data waardes in die gedeeltes. 'n Voorbeeld van 'n bedryfstelsel wat gebruik maak van segmentering is Multics[Pfl89].

Elke segment het 'n unieke naam. Kode- of data-items in 'n segment word dan geadresseer as 'n <naam,afset> paar, waar die naam die naam van die segment is wat die data-item bevat en die afset die plasing van die data-item in die segment is. Die afset is die plasing vanaf die begin van die segment. Die bedryfstelsel moet nou 'n tabel met segmentname en hul werklike addresse hou.

'n Gebruiker se program weet nie wat die werklike geheue-addresse is wat dit gebruik nie, en daar bestaan ook nie 'n manier om die werklike adresse te bepaal nie. Die sekerheidsvoordele hiervan is die volgende :

(a) Elke adresverwysing kan getoets word vir beskerming

(b) Verskillende beskermingsvlakke kan aan die baie verskillende klasse van data-items toegeken word

(c) Twee of meer gebruikers kan toegang tot 'n segment deel, maar met verskillende toegangsregte

(d) Dit is onmoontlik vir gebruikers om 'n adres of toegang tot 'n ongemagtigde segment te genereer.

3.6.7.Paginering

Paginering word op dieselfde prinsiep as segmentering gedoen, maar die gedeeltes waarin die program opgedeel word, is almal van gelyke grootte. Geheue word ook in dieselfde grootte eenhede opgedeel, en dié gedeeltes word pagina-rame genoem.

Elke adres word op dieselfde wyse bereken as met segmentering en word ook met 'n <bladsy,afset> veeltal bereik.

3.7. SERTIFISERING

Gebruikersertifisering word meestal deur 'n bedryfstelsel gedoen en dan baseer dit baie van sy beskerming op dit wat hy weet van 'n gebruiker van die stelsel. Die mees algemene sertifiseringsmeganisme is 'n wagwoord, 'n "woord" wat slegs bekend behoort te wees aan die gebruiker en die stelsel, want die menslike sy van die saak laat gewoonlik hierdie geheim uit [Pfl89].

Sertifisering is 'n aktiwiteit wat identiteit verifieer tussen of die een of beide van die entiteit betrokke in die sertifiseringsdialoog. Daar bestaan twee tipes sertifisering wat normaalweg in ag geneem word. Die eerste is magtiging van gebruikers wat aanteken op 'n stelsel en die tweede is sertifisering van rekenaars wat in 'n netwerk of verspreide omgewing werk[OSh91].

3.7.1. WAGWOORDE

Wagwoorde is wedersyds erkende kodewoorde wat veronderstel is om bekend te wees slegs aan die gebruiker en die stelsel waar enige van die twee partye die wagwoord kan kies. Die lengte en formaat kan ook verskil van stelsel tot stelsel. Die gebruik van 'n wagwoord is soos volg : 'n gebruiker sleutel 'n gedeelte van identifikasie in, soos 'n naam of 'n toegekende gebruikersID waar hierdie identifikasie beskikbaar mag wees aan die publiek of maklik wees geraai kan word omdat dit nie die werklike sekerheid van die stelsel bied nie. Die stelsel versoek dan die gebruiker vir 'n wagwoord. As die wagwoord die een op die lêer pas word die gebruiker gesertifiseer vir die stelsel. As die wagwoord faal, kon die gebruiker verkeerd getik het en word hy weer versoek vir 'n wagwoord.

Die stelsel het gewoonlik 'n wagwoordlys wat hy gebruik om die wagwoorde wat ingevoer word te valideer, d.w.s. die wagwoord is gewoonlik die eerste teiken wat aangeval word.

'n Veiliger manier om die wagwoordlys te beveilig is deur gebruik te maak van enkripsie. Twee algemeen gebruikte maniere om 'n wagwoordlys te enkripteer, is deur konvensionele enkripsie en een-rigting syfers. Konvensionele enkripsie beteken dat die hele wagwoordtabel geënkripteer word, of miskien net die wagwoordkolom. As 'n gebruiker se wagwoord ontvang word, word die gestoorde wagwoord gedekripteer en die twee word vergelyk met mekaar. Eenrigting enkripsie is 'n enkripsiefunksie waarvoor enkripsie relatief maklik en dekripsie relatief moeilik is. Die wagwoord in die wagwoordtabel word in 'n geënkripteerde vorm gestoor. As die gebruiker sy wagwoord insleutel, word dit ook geënkripteer en die geënkripteerde vorms word vergelyk.

GEVOLGTREKKING

Hoofstuk drie hanteer die verskillende meganismes wat gebruik kan word in die implementering van 'n sekerheidsmodel. Die meganismes wat bespreek word sluit onder andere die volgende in:

(a) Toegangsbeheerlyste

(b) Gidslyste

(c) Toegangsbeheermatrikse

(d) Eksklusief-uitsluitende Toegangsbeheermatrikse

(e) Vermoëgebaseerde toegangsbeheermeganismes

(f) Proseduregeoriënteerde toegangsbeheermeganismes.

Die bogenoemde omvat alle toegangsbeheermeganismes wat gebruik kan word. Die meganismes kan egter ook aangevul word met die gebruik van enkripsiemetodes en sertifiseringsmetodes, soos byvoorbeeld die gebruik van wagwoorde.

Toegangsbeheermeganismes word gewoonlik in die bedryfstelsel geïmplementeer en het tradisioneel nie gelet op die beskerming van die semantiese aspekte van data nie, daarom is daar verskeie beheermeganismes wat in databasisse ingebou word om verdere beheermaatreëls by te voeg by die bestaande. Die databasisbeheermeganismes bied veral goeie naam-afhanklike en konteks-afhanklike beheer. Meganismes wat in hierdie kategorie val, is byvoorbeeld die gebruik van sienings.

Geheuebeskermingstegnieke speel 'n belangrike rol in die implementering van onder andere die toegangsbeheermeganismes. Vermoëgebaseerde toegangsbeheermeganismes maak byvoorbeeld gebruik van segmentering in die implementering van die meganisme.

Die bespreking van sekerheid word hiermee afgehandel en daar word oorbeweeg na 'n bespreking van objekgeoriënteerde konsepte. Die kombinering van hierdie aspekte sal die grondslag lê vir die nuwe model.

---oOo---

HOOFSTUK 4

DIE OBJEKGEORIËNTEERDE PARADIGMA

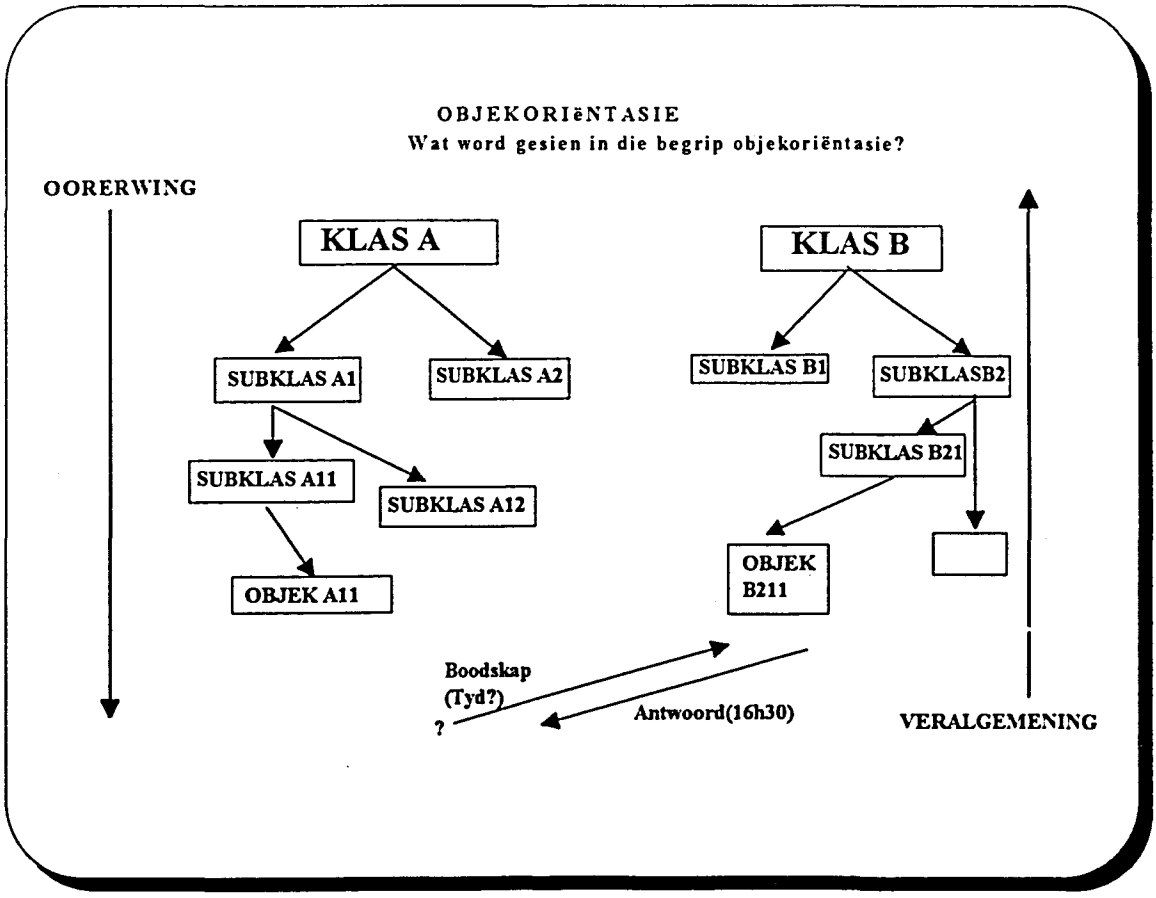
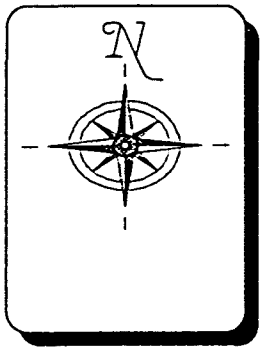


Fig 4.1. Die Objekgeoriënteerde Omgewing

4.1. INLEIDING.



Objekoriëntasie het vandag 'n belangrike rol, veral omdat dit verskeie voordele bied, onder andere die hergebruik van kode, die "verplaasliking" van verandering, die aanbied van ontwerpsbystand, uitbreibaarheid, aanpasbaarheid en laastens vinniger ontwikkelingsmoontlikhede. Die vier primêre eienskappe wat objekgeoriënteerde tale skei van ander kategorieë tale is oorerwing, dinamiese binding[Cox91], polimorfisme en dataversteking[Smi91].

Objekgeoriënteerde programmering is 'n kodeverpakkingstegniek - 'n tegniek wat die kodeverskaffer gebruik om funksionaliteit vir verbruikers te enkapsuleer[Cox91]. 'n Objekgeoriënteerde program kan gesien word as 'n versameling van beperkings op 'n interne toestand, met bevel-aksies wat die toestand kan verander[Hor92]. Dit kan ook soos volg gesien word : Dit is 'n ontwerps- en programmeringsmetode waar die dele van die ontwerp objekte is, wat gegroepeer word in klasse vir spesifikasie doeleindes, en waar daar afhanklikhede bestaan tussen die klasse wat gebruik word om spesialisering en veralgemeninge uit te druk[CAC90].

Objekoriëntasie kan die implementering van die voorgestelde model aansienlik vergemaklik, uitbou en deursigtig maak. Bruce Horn stel in sy artikel voor dat daar gebruik gemaak word van beperkingspatrone om die skryf van 'n objekgeoriënteerde stelsel nog verder te vergemaklik[Hor92]. Dit is dus belangrik dat die algemene konsepte van objekoriëntasie verstaan word. Indien u dus die objekgeoriënteerde paradigma onder die hande, het kan u hierdie hoofstuk weglaat. Dié hoofstuk behandel eers die algemene grondbeginsels van objekoriëntasie en daarna word die onderliggende besonderhede breedvoerig bespreek.

4.2. GRONDBEGINSELS

Die grondbeginsels van objekoriëntasie kan verdeel word in vyf komponente nl.,

(1) OBJEKTE

(2) METODES

(3) BOODSKAPPE

(4) KLASSE

(5) VOORKOMSTE

Hierdie komponente sal bespreek word deur die volgende vrae te beantwoord :

(a) WAT IS DIE KOMPONENT?

(b) HOE LYK DIE KOMPONENT?

(c) WAARVOOR WORD DIE KOMPONENT GEBRUIK?

(d) HOE WORD DIE KOMPONENT GEBRUIK?

(e) HOE REAGEER DIE KOMPONENT OF WAT IS DIE GEDRAG VAN DIE KOMPONENT?

Na afloop van die bespreking van die bogenoemde punte sal die onderliggende besonderhede bespreek word, soos bv. ABSTRAKSIE, POLIMORFISME, BINDING, ens.

4.2.1. OBJEKTE

WAT IS OBJEKTE ?

Daar bestaan 'n hele aantal definisies van wat objekte is, maar die volgende bied die duidelikste siening van objekte nl. :

'n OBJEK is 'n sagteware of programmatuur analoog van die werklike wêreld dinge of entiteite[Mul90]. Dit is selfbevattend, wat beteken dat dit beide data en kode bevat. Dit kan ook beskryf word as basiese uitvoertyd-entiteite[CAC90].

'n OBJEK is 'n voorkoms van 'n klas (sien klas en voorkoms),

Buite die rekenaaromgewing, is die betekenis van 'n objek: enigiets is met goed gedefinieerde grense of beperkinge.

'n OBJEK is iets psigies of fisies in die omgewing van denke, gevoel of aksie[Pin91],

'n OBJEK is 'n geënkapsuleerde abstraksie(sien abstraksie hieronder) wat 'n interne toestand besit, soos gegee deur 'n lys eienskappe met waardes wat uniek

tot die objek is. Die objek ken ook die lys van boodskappe waarop dit kan reageer en hoe dit sal reageer op elk[Pin91].

Daar bestaan ook Meta-objekte. Meta-objekte is objekte wat ander objekte verteenwoordig, d.w.s. dit besit nie die eienskappe van werklike objekte nie, maar beheer eerder die gedrag van die stelsel. Meta-objekte implementeer komplekse abstraksies, gebaseer op samewerkende fisiese objekte - waar die abstraksies gebaseer word op 'n hoë vlak siening van die stelsel se gedrag. 'n Meta objek word gevorm uit twee of meer unieke objekte wat self ook meta-objekte mag wees[Mul90].

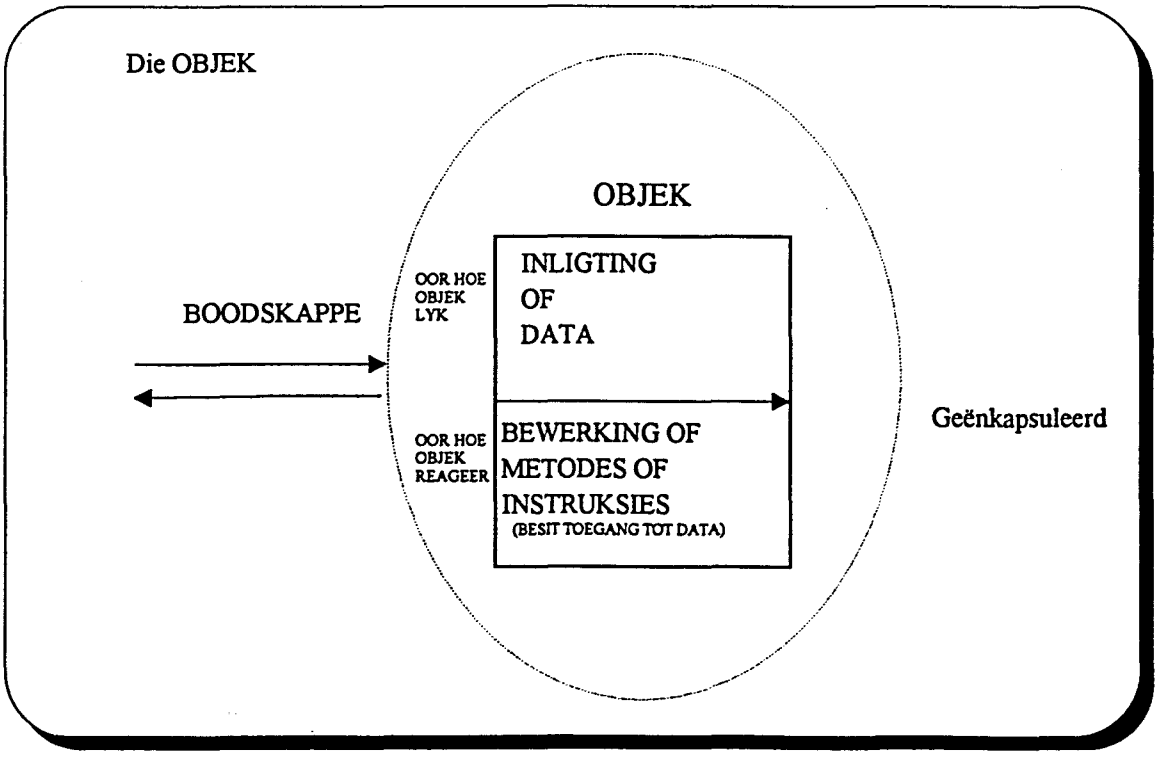


Fig 4.2. Die Objek

HOE LYK 'N OBJEK?

Objekte, soos gesien in figuur 4.2, is die manifestering van intelligente data. Dit bevat beide rou inligting en funksies wat daarop werk om sodoende betekenis aan

die inligting te gee. OBJEKTE besit dus inhoud en eienskappe (of funksies)[Smi91]. Dit bestaan uit twee gedeeltes

(1) Inligting en

(2) Bewerkings.

Objekte neem ruimte op in geheue en het 'n geassosieerde area soos 'n rekord in die tradisionele tale[CAC90]. Objekte steek die inligting weg, oftewel maak gebruik van enkapsulering. Inligtingversteking brei uit na data-abstraksie. Die groepering van bisse in die objek se geallokeerde geheueruimte bepaal die objek se toestand op enige oomblik[CAC90]. Die enigste manier om die inligting te verander is om die metodes in die objek te gebruik. Dus kan gesê word dat 'n objek beide toestand en gedrag enkapsuleer[CAC90].

Objekte verteenwoordig of gedra hulself net op een manier, nie op verskillende maniere nie, d.w.s. hulle is altyd konsekwent. D.w.s. ons kan altyd vertrou dat hulle hulself op 'n algemene uniforme wyse sal gedra.

HOE GEBRUIK ONS 'N OBJEK?

Die feit dat die interne toestand van die objek versteek word deur middel van enkapsulering, veroorsaak dat die objek slegs gebruik kan word deur versoeke daaraan te stuur. Indien hy die versoek aanvaar, aanvaar hy die verantwoordelikheid om die taak uit te voer. Na die uitvoering van die taak, stuur hy die antwoord wat gevra is, terug. Indien die versoek of taak nie 'n antwoord vereis nie, word die taak net uitgevoer.

'n Objek of data van 'n objek kan nie op 'n eksterne wyse verander word nie[Mul90]. Inligting wat versteek word in 'n objek, kan slegs verander word met behulp van die metodes gespesifiseer in die objek wat slegs weer gebruik kan word deur middel van boodskappe(sien boodskap), d.w.s om enige data te wysig in 'n objek of om dit te kry om 'n funksie uit te voer moet 'n versoek aan die objek

gestuur word wat dit vra om die aksie uit te voer[Mul90]. Die objekte word gewoonlik gevra om dinge te doen, vir of aan hulself[Smi91].

Enige voorkoms van 'n objek word beperk om verandering te maak aan sy eie private data[Mul90].

'n Objek bied dus 'n diens aan sy kliente (programme of gebruikers)[Wir90].

4.2.2. KLASSE

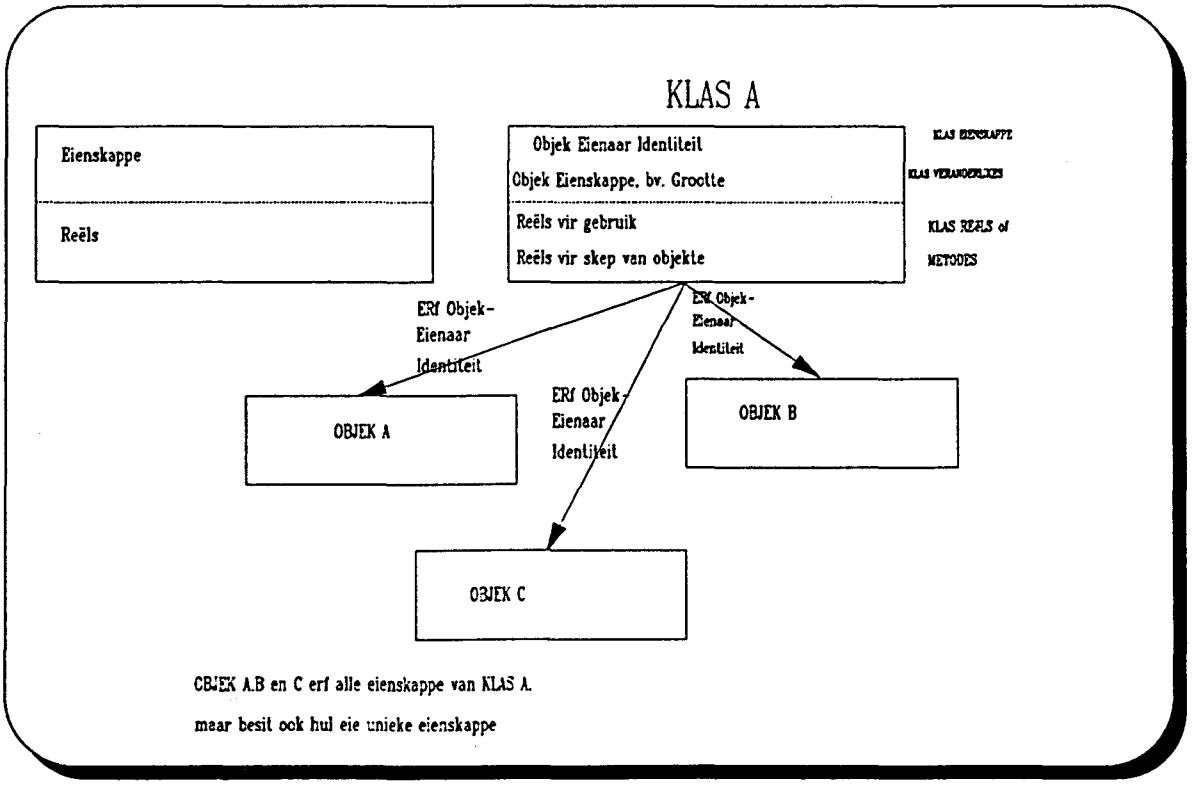


Fig 4.3. Die Klas

WAT IS 'N KLAS ?

Definisie van 'n klas :

(a) 'n Klas is 'n resep om 'n objek te bou[Smi91].

(b) Objekte word gedefinieer deur 'n KLAS wat gesien kan word as 'n bloudruk of sjabloon vir die skepping van spesifieke objekte wat hulle op dieselfde wyse gedra[Sav90].

(c) Klasse is ook OBJEKTE[Sav90], daarom bestaan daar 2 tipes metodes, nl. objekmetodes en klasmetodes.

(d) Klasse is nie soseer dinge as wat hulle reëls of sjablone is om dinge te maak nie[Mul90]. 'n Klas kan nie regtig iets doen nie, omdat dit nie 'n konkrete entiteit is nie, dit is eerder 'n bloudruk of sjabloon vir die konstruering van konkrete voorkomste self[Mul90].

(e) 'n Klas is 'n sjabloon vir die skep van 'n objek[Pin91].

(f) 'n Klas is die objek wat al die algemene eienskappe van elk van die elemente in 'n stelsel definieer, d.w.s. die eienskappe wat hulle in gemeen het. Dié objek, die klas, word dan aan die bo-punt van die hiërargie geplaas en in plaas van dat elke objek nou hierdie eienskappe bevat, word dit geërf vanaf die klas[Mul90]

(g) Die klas van 'n objek bepaal die volgende :

die naam vir watter tipe objek dit is,

die data geassosieer met die objek,

die funksies wat uitgevoer kan word op die objek

die klas waarvandaan die objek afgelei word[Sav90].

(h) Die objek klas definieer die data-strukture en -metodes vir die implementering van objekte as 'n private deel wat die objek se veranderlikes bevat en 'n ander deel wat gedeel word deur alle voorkomste van die klas[Cox91]. 'n Basiese verantwoordelikheid van die objek klas is die daarstelling van masjienerie om objekte te skep en te vernietig. Die gedeelde deel van sy fabriek word geërf deur alle ander fabrieksobjekte, sodat die allokeringsmasjienerie gedefinieer vir die objek fabriek geërf sal word deur alle ander fabrieksobjekte[Cox91].

'n Klas weet wat die veranderlikes is wat in die objek is en dit hou die metodes vir elk van die objekte[Smi91].

'n Klas is 'n familie van objekte van 'n spesifieke tipe[Sav90]. Objekte wat aan 'n klas behoort, word VOORKOMSTE genoem[Sav90]. 'n Klas mag verskeie voorkomste

besit, maar elke voorkoms het presies een klas. 'n Klas besit metodes en veranderlikes. Alle VOORKOMSTE van een klas het identiese metodes. Die klas-metode-voorkoms konsep is dus net 'n analogie van modules-prosedures-plaaslike veranderlikes[Sav90]. Klasmetodes word gebruik om nuwe tipes te definieer[Sav90].

Klasse word saam gegroepeer in 'n hiërargie wat strek vanaf baie algemene eienskappe wat van toepassing is op baie gespesialiseerde klasse tot by baie spesifieke eienskappe wat op 'n enkele klas van toepassing is[Mul90]. Dit wil sê, as ons klasse gebruik groepeer ons objekte saam volgens hul ooreenkomste[Mul90] om dan 'n klashiërargie te vorm. Dié **Klashiërargie** voorsien 'n raamwerk waarin abstraksies van dinge gedefinieer word. Die hoër vlakke in die hiërargie definieer abstrakte konsepte, algemene protokolle en gedeelde kode. Laer vlakke definieer konkrete implementerings. Dit is algemeen om soveel as 80% of meer van die kode in hoër vlakke van die hiërargie te besit[SMI91]. Die hiërargie word gebou op die volgende wyse :

Klasse kan SUBKLASSE besit, waar 'n SUBKLAS 'n klas is wat gebaseer word op 'n ander klas in die hiërargie en waar die hiërargie 'n verwante stel konsepte is wat werk vanaf die meer algemene na die meer spesifieke[Smi91]. Subklasse erf alles van sy superklas, bv. voorkomsveranderlikes, klasveranderlikes, metodes, ens. Die programmeerder skep 'n nuwe objek deur verskille tussen bestaande en die nuwe klasse te spesifiseer[Sav90].

Die klas wat 'n subklas bevat word 'n SUPERKLAS genoem[Sav90] (byvoorbeeld in Smalltalk).

Daar is net een klas wat geen subklas het nie, nl. die OBJEK.

'n SUBKLAS erf alles vanaf sy SUPERKLAS : voorkomsveranderlikes, klasveranderlikes, metodes[Sav90].

Klasveranderlikes is die veranderlikes wat gedeel word deur alle objekte in 'n klas, waar **voorkomsveranderlikes,** veranderlikes is wat gedupliseer word in elke bestaande objek. Klasveranderlikes is baie keer kontrolewaardes wat gebruik word op 'n enkele punt in die stelsel en mag in hierdie gevalle versteek wees in die objek self deur dit te verplaas as plaaslike statiese veranderlikes in die lede funksies wat hulle gebruik[Mul90].

HOE LYK 'N KLAS?

'n Klas sluit 'n objek se beskrywing in, asook 'n naam vir die tipe objek, 'n lys van eienskappe en 'n lys van boodskappe met ooreenstemmende metodes waarop 'n objek van die klas kan reageer[Pin91]. Tussen die boodskappe in die klasbeskrywing is die wat gebruik word om die voorkomste van die klas te skep. Dié voorkomsskeppings word aan die klasnaam gestuur[Pin91].

'n Klas is die definisie van 'n objek. 'n Klas bestaan uit die lys van data wat in elke voorkoms van die objek is, die metodes wat sal reageer op die data en die metodes wat aan die klas self behoort[Smi91].

Klasse bevat klasveranderlikes en klasmetodes. Klasveranderlikes is veranderlikes wat tot 'n klas behoort. Daarna word verwys deur 'n voorkomsmetode of 'n klasmetode. Dit is dus globaal tot die klas en sy subklasse[Smi91]. Klasmetodes behoort tot die klas self en word tipies gebruik om nuwe voorkomste te allokeer[Smi91]. Klasmetodes kan dieselfde name as voorkomsmetodes besit.

HOE GEBRUIK ONS 'N KLAS?

Die programmeerder skep 'n nuwe objek deur verskille tussen bestaande en die nuwe klasse te spesifiseer. Die metodes van die klas word gebruik om nuwe objekte te skep deur aan die klas boodskappe te stuur om die nuwe objekte aan te vra.

Sommige stelsels, bv. Smalltalk, besit uitvoertyd-klas-objekte of meta-klas-objekte[CAC90]. Dit beteken dat vir elke klas in die stelsel daar 'n ooreenstemmende objek bestaan wat inligting stoor oor die klas as geheel en wat die klas vlak bewerkings implementeer. Sulke objekte is nuttig vir die stoor of berging van dinamiese inligting oor die tipe, soos bv. die aantal huidige voorkomste van die tipe, en voorsien 'n elegante manier om bewerkings soos klaskonstrueerder en -dekonstrueerder om die globale inligting wat spesifiek tot die klas is, te hou[CAC90].

4.2.3. METODES

WAT IS METODES ?

Metodes word soms bewerkings, instruksies, programme, prosedures of aksies genoem. Dit vertel hoe objekte reageer en wat om te doen as iets met die objek gebeur. Die entiteite in die voorgestelde model sal ook sekere reaksies besit - hulle sal bv. lêers wil lees, die stelsel wil gebruik, data wil bywerk, ens.

Aksies of bewerkings hanteer die evaluering of verandering van inligting in 'n objek. Dit kan ook nuwe inligting skep, oftewel nuwe objekte. 'n Metode of bewerking bestaan uit 'n aantal stappe.

Definisie : 'n Metode is 'n instruksie wat spesifiseer hoe 'n objek iets doen, oftewel hoe dit werk[Smi91]. Metodes vertel ook hoe aksies uitgevoer moet word[Smi91]. 'n Subroetine of prosedure behoort tot 'n klas en word uitgevoer deur 'n boodskap te stuur. As 'n metode gevind word, word dit uitgevoer net soos wat 'n subroetine uitgevoer word in die konvensionele tale[Smi91]. 'n Metode is 'n lys van instruksies met volledige besonderhede wat definieer hoe 'n objek reageer op 'n spesifieke boodskap[Pin91]. 'n Metode bestaan tipies uit uitdrukkings wat die funksie van boodskappe uitvoer wat na die objekte gestuur word. Elke boodskap in 'n klas moet 'n ooreenstemmende metode bevat[Pin91].

Definisie : 'n Metode is 'n subroetine of prosedure wat behoort tot 'n klas en wat uitgevoer word deur - 'n boodskap wat gestuur was, te beantwoord. 'n Metode, as dit gevind kan word, word uitgevoer op dieselfde wyse as wat 'n subroetine in 'n konvensionele taal uitgevoer is.

Metodes benodig soms inligting of invoere wat dit moontlik maak om die nodige antwoord te verskaf of aksie uit te voer. Die invoere na 'n metodes is ook veranderlikes, maar nie die veranderlikes wat definieer wat 'n objek is nie; hulle hou eintlik net die inligting wat die metode benodig.

Indien subklasse gedefinieer word is alle metodes nie noodwendig in die superklas nie, maar kan ook in die subklasse lê.

Metodes wat geërf word vanaf 'n ouerklas kan uitgevoer word deur die kind klas asof hulle die kind se metodes was. 'n Kind kan 'n ouer se metodes vervang met dieselfde naam waar die vervangingsmetode uitgevoer word in plaas van een wat behoort aan die ouer.

Metodes is net prosedures wat geïnisieer word as 'n boodskap ontvang word vanaf die buitewêreld. Alle voorkomste van 'n klas het identiese metodes[Sav90].

HOE LYK 'N METODE?

'n Metode bestaan uit 'n aantal instruksies of uitdrukkings wat die funksies definieer waarop 'n objek sal reageer, of wat die aksies wat die objek sal uitvoer definieer.

HOE GEBRUIK ONS 'N METODE?

'n Metode word uitgevoer deur aan die objek 'n boodskap te stuur. Die boodskap identifiseer 'n aksie wat die objek moet uitvoer en deur laat of vroeë binding word die metode gekoppel aan die boodskap, uitgevoer en die aksie is voltooi.

DIE SOEKTOG VIR 'N GESKIKTE METODE

Metodes moet gevind word voordat hulle uitgevoer kan word. Met oorerwing, kan ons meervoudige metodes met dieselfde naam besit. Die tegniek om die regte klas en metode te vind is baie belangrik : Daar is twee stappe om 'n metode te vind :
(a) bepaal vanaf watter klas die soektog sal begin, en
(b) vind die metode in hierdie klas of sy ouers[Smi91].

Die klas waarvandaan die soektog begin word, is gewoonlik die klas van die voorkoms aan wie 'n boodskap gestuur word. As 'n begin klas eers gevind is gaan die soektog verder in daardie klas. As dit nie gebind was nie, word daar na die voorouers van hierdie klas gekyk[Smi91].

4.2.4. BOODSKAPPE

WAT IS BOODSKAPPE ?

Boodskappe in objekgeoriënteerde programmering lyk of kan vergelyk word met die roep van subroetines in die tradisionele programmeringstale. 'n Boodskap is 'sinchroon of gelyktydig' wat beteken dat die metode wat deur die gestuurde boodskap aangevra is, nie uitgevoer word totdat die metode wat die boodskap uitvoer voltooi is nie[Smi91]. Boodskappe word gebruik om die toestand van objekte te verander, deur middel van bevel aksies wat geënkodeer word in metodes[Hor92].

Daar bestaan verskeie tipes boodskappe :

Unêre boodskappe - Boodskappe met geen invoere. Dit word ook gedefinieer as 'n boodskap met geen argumente. Die boodskappe bestaan uit twee gedeeltes, naamlik :

'n Naam van die objek wat die boodskap ontvang en die naam van die boodskap[Mul90].

Sleutelwoordboodskappe - Boodskappe met 'n sleutelwoord as invoer,
(bv. AnimasieObjek : Vertoon Stadig en
AnimasieObjek : Vertoon Vinnig)

Binêre boodskappe - Boodskappe met twee objekte as parameters,

Klasboodskappe - Boodskappe aan klasse

Metodes kan slegs bereik word of gebruik word deur aan dit 'n boodskap te stuur.

Definisie Boodskap : Die definisie boodskap word aan 'n voorkoms gestuur om 'n metode uit te voer. Die Boodskap wat gestuur word het dieselfde naam as die metode wat uitgevoer word[Smi91].

Meta boodskappe groepeer 'n reeks boodskappe saam in objekgeoriënteerde programmering[Mul90]. Boodskappe is baie nou verbind of gekoppel aan die objek waarmee dit geassosieer word[Mul90].

HOE LYK 'N BOODSKAP?

Dit is dit wat aan 'n voorkoms gestuur word om 'n metode uit te voer. 'n Boodskap het dieselfde naam as die metode wat uitgevoer moet word[Smi91].

Boodskappe is baie nou verbind of gekoppel aan die objek met wie hul geassosieer word[Mul90].

Boodskappe is nie 'n gelyktydigheidsmeganisme nie, maar eerder 'n modulariteitsmeganisme. Die stuur van boodskappe skep die enkapsulering van data en prosedures en prosedures wat 'n objek genoem word[Cox91].

'n Boodskap word verteenwoordig deur 'n identifiseerder, 'n spesiale simbool of kombinasie van identifiseerders wat 'n aksie wat genoem moet word deur die objek impliseer. Boodskappe mag eenvoudig wees of mag parameters insluit wat affekteer hoe die ontvangende objek reageer. Die manier waarop 'n objek reageer op 'n boodskap mag ook geaffekteer word deur die waardes van sy eienskappe[Pin91].

HOE GEBRUIK ONS 'N BOODSKAP?

Kliënte versoek dienste van 'n objek en omdat objekte geënkapsuleer is sodat die kliënt nie die interne data of struktuur van die objek kan sien nie, stuur die kliënt aan die objek 'n boodskap[Wir90]. 'n Versoek identifiseer die versoekte diens, sowel as die objekte wat die diens moet uitvoer. Dié objekte kan ondubbelsinnig geïdentifiseer word en is betroubaar. Sulke versoeke mag generies wees, d.w.s. die kliënt kan dieselfde versoek rig aan verskillende tipes objekte wat dieselfde tipe diens verrig[Wir90]. Wanneer 'n versoek uitgereik word, bepaal 'n seleksie proses die werklike kode of diens wat uitgevoer moet word. Meer as een objek kan saamwerk as antwoord of respons of die versoek.

Boodskappe word aan objekte gestuur om hulle te vra om sekere funksies of aksies uit te voer. Boodskappe kan ook aan klasse gestuur word wat in der waarheid sê dat klasse ook metodes mag besit. Gewoonlik skep hierdie metodes nuwe voorkomste van die klas[Smi91]. Die "nuwe" boodskap is net 'n stelsel-gedefinieerde boodskap aan 'n klas.

Indien 'n boodskap na 'n objek gestuur was, word dit die ontvanger se verantwoordelikheid om korrek te reageer[Mul90]. Dieselfde boodskap kan verskillende antwoorde verkry vanaf verskillende objekte.

Die feit dat 'n objek of data van 'n objek nie op 'n eksterne wyse verander kan word nie, maak dit noodsaaklik dat daar 'n manier bestaan om die objek te kry om die aksie of funksie uit te voer wat die data van die objek of die objek self verander. Dit word

boodskappe genoem en word aan die objek gestuur om dit te vra om die aksie uit te voer[Mul90].

4.2.5. VOORKOMS

Definisie:

(a)'n Voorkoms is 'n objek wat data hou, asook 'n verwysingspunt na sy klas[Smi91].

(b)Die data gedefinieer deur 'n klas, asook 'n verwysing na die klas. 'n Voorkoms hou die data; die klas hou die kode; die klas waarna verwys word in die voorkoms verseker dat boodskappe wat gestuur word die toepaslike metode vind. Voorkomste word ook objekte genoem[Smi91].

(c)'n Metode in 'n klas wat uitgevoer word as 'n boodskap gestuur word aan die voorkoms[Smi91].

Definisie: Voorkomsveranderlike

Die data in 'n voorkoms, gelys en benoem in die gedefinieerde klas, word 'n lys of voorkomsveranderlike genoem.

4.2.6. BINDING

OPERATOROORLAAIING : Die vermoë om operators gesamentlik met operandes van verskillende tipes tydens program uitvoering te gebruik. Dinamiese binding word hiervoor gebruik[Mul90].

WAT IS BINDING ?

Binding is die proses om funksionaliteit van verskillende verskaffers te integreer in 'n verbruikerskode[Cox91]. Dit is die proses waarvolgens operators en operandes van

potensieel baie verskillende tipes, gepubliseer kan word deur verskaffers, en gebruik kan word deur verbruikers[Cox91].

Die proses om 'n tipe aan die veranderlike naam te heg word binding genoem. Daar word aan elke veranderlike naam 'n plek en 'n tipe in die geheue geallokeer. Vroeë binding word gewoonlik in tradisionele tale gebruik en geskied gewoonlik met vertaaltyd, terwyl laat binding in objekgeoriënteerde tale gebruik word en geskied as die tipe vereis word, dit wil sê tydens programuitvoertyd. Dit speel 'n belangrike rol in operatoroorlaaiing. Laat binding is die vermoë van 'n taal om dieselfde sintaktiese operator op verskillende datatipes toe te pas[Sav90]. Laat binding geskied later as vertaaltyd, gewoonlik as die program alreeds uitgevoer word[Cox91]. Vroeë binding is ook wanneer die verbruiker se kode so vertaal word dat die gebruiker en sy hulpmiddels die verantwoordelikheid dra vir binding[Cox91].

Vroeë binding werk in 'n geslote heelal waarin alle potensiële interaksies tussen die dele van die omgewing verklaar kan word wanneer hierdie gedeeltes geskep word deur die vertaler. Laat binding word essensieel in 'n oop heelal gebruik as die gedeeltes wat in wisselwerking is, onbekend is aan die program wat uitgevoer word[Cox91].

HOE GEBRUIK ONS BINDING?

Laat binding word gebruik deur die verklaring van tipes programmaties te doen tydens programuitvoering, terwyl vroeë binding deur die gebruiker voor die uitvoering van die program in die program geprogrammeer word.

4.2.7. ABSTRAKTE KLASSE

WAT IS ABSTRAKTE KLASSE ?

Definisie :

'n Abstrakte klas is 'n ouerklas of superklas wat bestaan uit 'n protokol alleen. Dit is nie bedoel om voorkomste te bevat nie. Dit word bedoel dat die abstrakte klas

reageer as 'n model vir 'n subklas of subklasse wat die minimum protokol definieer wat hulle moet ondersteun. 'n Protokol is 'n lys van metodes wat deur 'n klas ondersteun word of 'n gelyktydige deelversameling van die lys van metodes ondersteun deur 'n klas[Smi91]. Abstrakte superklasse word gebruik om versamelingsmetodes, genoem protokolle, te implementeer. Abstrakte superklasse is gewoonlik onvolledig en enige voorkoms wat gevorm word uit die abstrakte klas sal nutteloos wees.

Versamelings is ook abstrakte klasse, maar wat is versamelings? Versamelings is die vereniging van eenderse objekte. Alle versamelings moet reageer op 'n sekere versameling boodskappe om te kwalifiseer as versameling.

HOE LYK 'N ABSTRAKTE KLAS?

Metodes word gedefinieer in 'n abstrakte klas wat werk vir al sy subklasse, d.w.s. hierdie gedeelte lyk dieselfde as vir 'n gewone klas.

HOE GEBRUIK ONS 'N ABSTRAKTE KLAS?

Abstrakte klasse word gewoonlik gebruik as kontrolemeganismes in 'n stelsel.

4.2.8. OORERWING

Definisie :
Oorerwing is die proses van verkryging van karakteristieke van 'n ouer in 'n hiërargie. Karakteristieke wat geërf kan word in Objekgeoriënteerde programmering is voorkomsveranderlikes, voorkomsmetodes, klasveranderlikes en klasmetodes[Smi91]. Oorerwing is ook 'n hulpmiddel om herbruikbare klasse te organiseer, te bou en te gebruik. Sonder oorerwing sal elke klas 'n vrystaande eenheid wees wat elk van die grond af op ontwikkel is. Verskillende klasse sal geen verwantskap met mekaar besit nie, omdat die ontwikkeling van elke metode voorsien word soos hoe ookal hy dit kies[Cox91]. Oorerwing verbind konsepte in 'n verwante geheel[Pin91], sodat as 'n

hoër vlak konsep verander, die verandering van toepassing is. Dit kan ook gesien word as 'n verhouding tussen klasse wat toelaat dat die definisie en implementering van een klas gebaseer word op die van 'n reeds bestaande klas[CAC90].

Oorerwing is die daad om 'n besitting, kondisie of ruiling uit vorige generasies te verkry. In objekgeoriënteerde probleemoplossings erf een tipe objek eienskappe wat 'n ander tipe objek karakteriseer. Die feit dat die eienskappe van objekte deur klasbeskrywings gegee word, impliseer dat dit in 'n hiërargie van klasse is, waar een klas 'n subklas van 'n ander, ouerklas is. Objekte wat voorkomste is van die subklas beskrywing, besit sowel geërfde eienskappe gegee in die ouerklas asook eienskappe gegee in voorouerklasse. Voorkomste van 'n subklas verteenwoordig 'n spesialisering van voorkomste beskryf deur 'n ouerklas. Die subklas voorkoms het al die eienskappe gegee deur die ouerklas plus addisionele attribute[Pin91].

Oorerwing is wanneer 'n afgeleide klas data en funksionele vermoëns van sy basisklas gebruik[Mul90]. Oorerwing spruit voort uit die feit dat objekte beide data en funksies wat op die data werk bevat[Mul90]. Oorerwing is net 'n manier om weg te beweeg vanaf die algemene na die meer gespesialiseerde vlakke in die hiërargie[Mul90].

Oorerwing word baie keer gebruik om abstraksie en struktuur teenwoordig in 'n toepassingsdomein te weerspieël. 'n Voorbeeld van 'n algemene toepassingsdomein is grafika[CAC90]. Die eintlike voordeel van die oorerwingsmeganisme is dat dit die programmeerder toelaat om 'n klas wees te gebruik wat amper, maar nie heeltemal, op so 'n manier gevorm is dat dit nie onwenslike newe-effekte in die res van die klas voorstel nie.

Streng oorerwing vereis dat afgeleide klasse aanpasbaar moet wees by die basisklasse. Nie-streng oorerwing laat toe dat lede funksies van 'n basisklas willekeurig oorskryf word[CAC90]. Die groot voordeel van streng oorerwing is dat dit gebruik kan word om tipe abstraksies te implementeer[CAC90]. Die laasgenoemde laat toe dat 'n program objekte van 'n klas en enige klasse daaruit afgelei uniform hanteer word. Dit laat ook toe dat die versameling klasse uitgebrei kan word sonder om die toepassings

wat ontwikkel is uit die basisklas, te verander[CAC90]. Tipe abstraksies het die nadeel dat dit beperk wat gedoen kan word. Nie-streng oorerwing is meer buigbaar, maar maak dit moeiliker om korrektheid te verseker as daar gebruik gemaak word van inkrementele verandering deur afleiding[CAC90].

4.2.10 ENKAPSULERING

Enkapsulering het drie moontlike betekenisse : (a) Die afdwinging van abstraksiegrense, (b) die aksie van integrering van eksterne komponente in die stelsel in en (c) die meganisme om toegang tot diens deur verskillende gebruikers te beheer[Wir90].

Enkapsulering is die daad van enkapsulasie/versteking. Die resultaat van enkapsulasie is 'n entiteit met definitiewe perke of grense, 'n goed gedefinieerde koppelvlak, en 'n beskermde interne voorstelling. Die integriteit van enkapsulasie hang van fasette van die onderliggende taal af. In objekoriëntasie is die eenheid van enkapsulasie die objek[Pin91].

Enkapsulering is die proses waar eienskappe van 'n program versteek word binne individuele objekte wat versprei is deur die stelsel[Mul90]. Enkapsulering is ook die grondslag of basis van die algehele objekgeoriënteerde benadering. Sy invoeging beperk die effekte van verandering deur 'n muur van kode om elke data gedeelte te plaas[Cox91]. Alle toegang tot data word hanteer deur prosedures wat daar geplaas is om toegang tot die data te bemiddel. Geënkapsuleerde operande word objekte genoem[Cox91].

Definisie : Enkapsulering : Hoe 'n objek sy aksies implementeer en hoe sy interne data gerangskik is, word versteek binne 'n proseduredop wat alle toegang tot die objek bemiddel[Cox91].

4.2.11 POLIMORFISME

Polimorfisme word gedefinieer as die kwaliteit of toestand om in staat te wees om verskillende vorms aan te neem[Pin91]. In objekgeoriënteerde probleemoplossing mag polimorfisme of bewerking polimorfisme wat verteenwoordig word deur boodskappe met dieselfde naam, gestuur word aan verskillende objekte, waar elk op sy eie wyse reageer[Pin91].

Die eerste eienskap van polimorfisme is die oorlaaiing van boodskapidentifiseerders op operatore. Polimorfisme word verder ondersteun deur die binding van 'n spesifieke metode op 'n boodskapidentifiseerder gedurende die uitvoering van 'n stelsel[Pin91].

Polimorfismes beteken net dat daar meer as een metode in verskillende klasse bestaan met dieselfde naam[Smi91].

4.2.12 ABSTRAKSIES

'n Abstraksie het konsepsuele eerder as konkrete bestaan. Dit verteenwoordig idees, konsepte en algemene eienskappe sonder om aandag te skenk aan besonderhede, dit beteken sonder om aandag te gee aan die implementeringsbesonderhede[Pin91].

In objekoriëntasie is beide objekte en boodskappe abstraksies. Elk verteenwoordig 'n konsepsuele komponent van die probleemoplossing met die vermoë vir verskeie onderliggende lae van addisionele abstraksies[Pin91].

Inligtingversteking brei uit na data-abstraksie[Smi91]. Inligtingversteking kan gesimuleer word tot 'n mate, maar ander gewensde objekgeoriënteerde taaleienskappe nie.

4.3. VOORDELE WAT DIE OBJEKGEORIËNTEERDE PARADIGMA BIED VIR DIE BOU VAN SEKERHEIDSMODELLE

KODE HERGEBRUIK

'n Bestaande kode is baie meer toepaslik om weer gebruik te word as 'n nuwe kode, want dit is reeds bewys dat dit korrekte resultate lewer. Dit is nuttig en prakties om 'n kode een keer te skryf en om dit dan oor en oor te kan gebruik[Smi91].

VERPLAASLIKING VAN VERANDERING

Veranderinge is gewoonlik plaaslik tot 'n klas[Smi91] in objekgeoriënteerde programmering. Verplaasliking van verandering kom gewoonlik voor, a.g.v. dataversteking en polimorfisme[Smi91]. Dit is selfs geïsoleer tot 'n nuwe subklas eerder as wat dit verspreid is oor die hele toepassing.

ONTWERPSBYSTAND

Die hiërargie wat klasse vorm, forseer 'n beskouing van die ontwerp van die stelsel, voor kodering kan plaasvind. Objekte word geskryf as spesialisering van ander objekte. Die ontwerp van 'n program vereis dinkwerk oor die objekte en die abstrakte konsepte of algemene dinge waarvan die objekte spesialiserings is, d.w.s. die klashiërargie is een van die eerste dinge wat ontwerp word[Smi91].

UITBREIBAARHEID

Enige objek of klas kan uitgebrei of deursigtig verander word in 'n stelsel, d.w.s. dit kan verander word terwyl die ander objekte of klasse dit nie eens sal sien of daardeur beïnvloed sal word nie.

VINNIGER ONTWIKKELING

Die ontwikkelingsiklus in Objekgeoriënteerde programmering (OOP) kan verkort word, omdat bestaande kode makliker hergebruik kan word en omdat spesifikasies baie gereeld verander en dit nie 'n probleem is om die stelsel dan te verander nie.

GEVOLGTREKKING

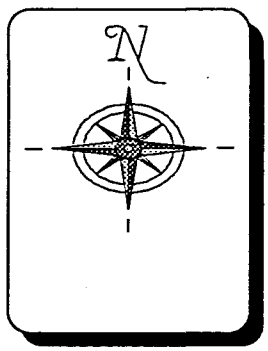
Hoofstuk vier hanteer die objekgeoriënteerde paradigma deur antwoorde op die vrae "Hoe","Wat", en "Wanneer" rakende die elemente van objekoriëntasie te verskaf. Die gebruik van binding, polimorfisme en oorerwing word ook behandel. Die feit dat rekenaar stelsels die beste beskerm word deur 'n sekerheidstelsel van dieselfde tipe, maak dit noodsaaklik dat 'n sekerheidstelsel ontwikkel word vir objekgeoriënteerde omgewings. Sekere eienskappe van objekoriëntasie veroorsaak egter nuwe probleme in die sekerheidsveld wat oorbrug moet word, en die enigste manier om hierdie probleme te oorbrug is om oor 'n grondige kennis te beskik van objekgeoriënteerde konsepte. Oorerwing, as eienskap van objekoriëntasie, veroorsaak opsigself heelwat nuwe hindernisse wat oorbrug moet word. Die volgende hoofstuk poog om riglyne saam te stel vir die ontwikkeling van só 'n nuwe sekerheidstelsel met die inagneming van al die probleme van objekoriëntasie.

---oOo---

HOOFSTUK 5

RIGLYNE VIR DIE BOU VAN 'N OBJEKGEORIËNTEERDE SEKERHEIDS MODEL.

Die vorige hoofstukke was gemik daarop om 'n algemene agtergrond aan te bied van die areas, objekoriëntasie en sekerheid. Die twee velde lê die grondslag waarop die DISMOD sekerheidsmodel gebou is. Nuwe sekerheidsmodelle word egter elke dag gekonstrueer om die nuwe programmerings- en ontwerpsvelde wat ontwikkel word, te ondersteun. Dit is raadsaam om dus te kyk na die stelsels wat in die verlede ontwikkel is maar ook die wat in die hede gebruik word, indien 'n nuwe sekerheidsmodel gebou word.



Daar word in hierdie hoofstuk gepoog om riglyne saam te stel waarop 'n nuwe model ontwikkel kan word. Dit is egter moeilik om die riglyne vir die algehele rekenaaromgewing wat vandag bestaan te ontwikkel, daarom word daar in hierdie hoofstuk slegs riglyne aangebied tot die bou van 'n diskresionêre sekerheidsmodel wat gebruik maak van objekgeoriënteerde konsepte.

5.1.INLEIDING

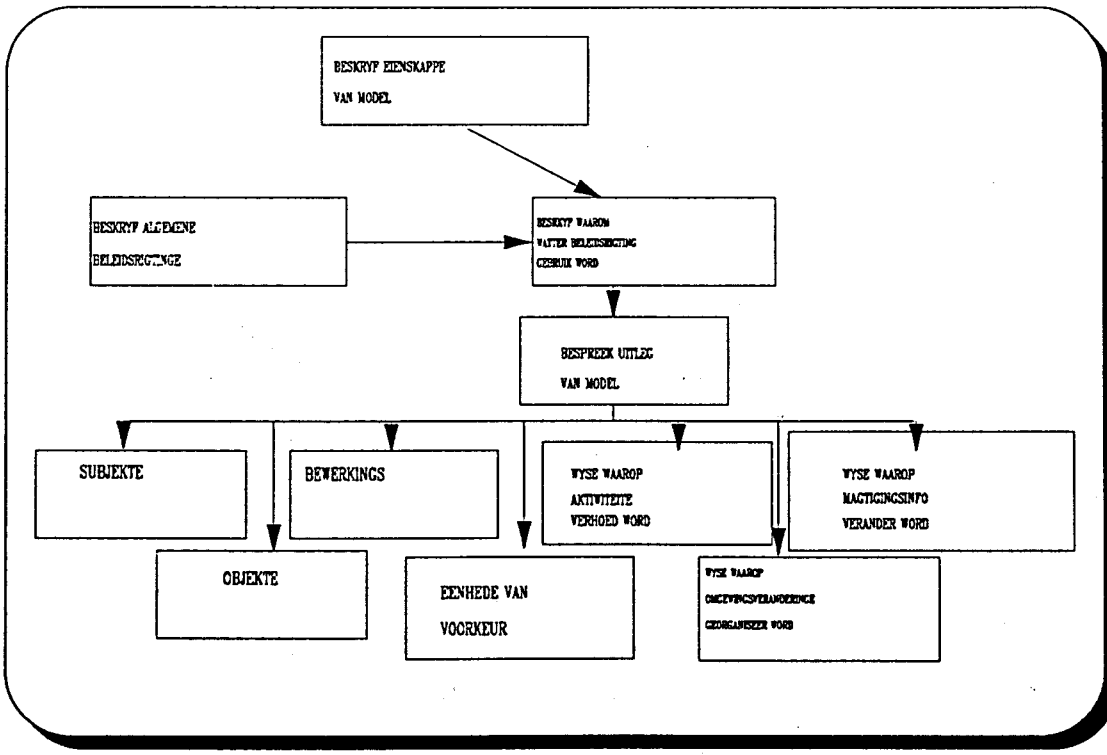


Fig 5.1. Die uiteensetting van die hoofstuk.

In hierdie hoofstuk sal riglyne saamgestel word wat gebruik word in die bou van 'n objekgeoriënteerde diskresionêre sekerheidsmodel. Daar sal verwys word na bestaande modelle (a) om gebruike in die praktyk uit te wys, en (b) om voorgestelde gebruike wat van toepassing is op verskillende situasies en omstandigdhede te illustreer. Daar word gebruik gemaak van 'n verskeidenheid van toegangsmetodes en toegangsbeheerstelsels in die verskillende modelle en elk van hierdie metodes speel 'n belangrike rol in die samestelling van 'n sekerheidsmodel. Dit is noodsaaklik om die grondbeginsels van objekgeoriënteerde sekerheidsmodelle te verstaan om sodoende 'n effektiewe model te kan ontwikkel.

Tradisionele sekerheidsmodelle het 'n tekort gehad aan semantiese uitdrukbaarheid. Dit het byvoorbeeld nie altyd voorsiening gemaak vir die beskerming van die betekenis van die data nie maar eerder vir die beskerming van die data self. Daar was ook nie voldoende beskerming vir komplekse hiërargieë en skakels tussen data elemente nie.

Mettertyd het daar egter verskillende nuwe modelle ontstaan om hierdie probleem die hoof te bied.

Dit wil voorkom asof die objekoriëntasie die rekenaarveld is waar daar meestal vandag nog baie oplossings gesoek moet word vir beskermingsvraagstukke. Ander voorbeelde van bestaande rekenaarstelsels waarvoor daar nog nie voldoende beskerming gegee word nie, is kantooroutomatiseringspakkette, kennisvoorstelling, objekgeoriënteerde grafiese pakkette. Die modelle wat gebruik word in hierdie hoofstuk poog almal om hierdie tekortkominge te bowe te kom. Die modelle wat gebruik gaan word in die bespreking sluit die volgende in: **DAMOKLES**(Discretionary Access Control in Structurally Object-oriented Database Systems), **DISCO**(A Discretionary Security Model for Object-oriented Databases), **EIUDM**(an Extended Universal Instance Datamodel), Model for **NEXT GENERATION DATABASES, SODA**(a Secure Object-oriented Database System), en Demurjian, et al. se '**REQUIREMENTS, CAPABIITIES AND FUNCTIONALITIES OF USER-ROLE BASED SECURITY FOR AN OBJECT-ORIENTED DESIGN MODEL**'.

Die uitleg van die hoofstuk is soos volg :

'n Model word meestal gebou deur eerstens die eienskappe van die model uiteen te sit. Dié eienskappe is baie keer geskoei op 'n tekortkoming in ander modelle wat oorbrug moet word. Afdeling een sal kyk na hoe die onderskeie modelle se eienskappe uiteengesit word, en daar sal ook verwys word na die rede vir hierdie uiteensetting. Nadat die eienskappe van die model uiteengesit is, word die sekerheidsbeleid van die maatskappy of die model uiteengesit. 'n Sekerheidsbeleid speel 'n belangrike rol in die eienskappe van 'n model (Afdeling een) en daar is omstandigdhede waar die twee nie van mekaar geskei kan word nie. In afdeling twee word daar terugverwys na afdeling 1. Die tweede afdeling bespreek hierdie beleidrigtings en voorbeelde sal gegee word waar die beleidrigtings gebruik word in die onderskeie modelle. Die beleidsrigting van 'n maatskappy moet deur die sekerheidsmodel ondersteun word[Lar90], of 'n maatskappy moet 'n beleid vir hierdie doel saamstel. Nadat die

laasgenoemde besluite geneem is, kan die model se struktuur uiteengesit word, waaronder die elemente van die model. In hierdie afdeling sal ook aandag geskenk word aan die metodes wat gebruik word om onveilige of skadelike aktiwiteite te verhoed. Voorbeelde van sulke metodes is byvoorbeeld die volgende:

(a) toepassing van toegangsbeheer, of
(b)die stel van reëls wat hanteer hoe die omgewingsveranderinge georganiseer sal word en
(c) Reëls wat definieer hoe die magtigingsinligting verander sal word, bv. deur gebruik te maak van 'n stelselsekerheidsbeampte of 'n betroubare rekenbasis.

Die laasgenoemde metode impliseer dat 'n reeks stappe gevolg sal moet word :
(A) Die Struktuur vir magtigingreëls moet saamgestel word[Lar90].
(B) 'n Raamwerk vir magtigingsbestuur moet saamgestel word[Lar90]
(C) 'n Navraag evalueringsalgoritme kan saamgestel word.
Die bogenoemde sal in afdeling drie uiteengesit word met behulp van voorbeelde.

Afdeling een sal vervolgens die uiteensetting van die eienskappe van 'n model hanteer.

5.2. DIE UITEENSETTING VAN 'N MODEL

Die uiteensetting van 'n model is een van die eerste stappe wat gevolg word in die bou van 'n nuwe model. In hierdie stap word die doel en eienskappe van 'n model voorgelê aan die leser. Die doel en eienskappe van 'n model verkoop gewoonlik die model. Afdeling 5.2. sal vervolgens so 'n uiteensetting bespreek, met die doel om (a) te wys hoe dit gedoen word en (b) om die eienskappe van verskeie reeds bestaande modelle uit te lig. Dit bied die geleentheid om goeie eienskappe voort te sit in die bou van die nuwe model, maar ook om swak eienskappe te vermy.

In die uiteensetting van 'n model word die volgende vrae gevra:

(a) *Wat is die doel van die model?*
(b) *Watter basisdatamodel gaan gebruik word om die model te bou,* bv. 'n semantiese datamodel of 'n objekgeoriënteerde datamodel ens.
(c) *Is die model konsekwent met die datamodel wat ondersteun word?,*
(d) *Watter karaktertrekke moet die model bevat?.*
(e) *Watter tipe beleid* (algemene sekerheid, maatskappy beleid, nuwe tegnologie beleid) *moet deur hierdie model ondersteun word?*

DOEL

'n Model word meestal saamgestel met 'n oorhoofse doel in die oog. Beskou bv. die ***Private Toegangskanaal[Dol93].*** Dié model se doel is om dieselfde vlak of grein van objekgebaseerde sekerheid te voorsien as die vermoëgebaseerde sekerheid skema. Dit voorsien 'n generiese beskermingsmeganisme wat toegepas kan word op enige objek en nie afhanklik is van die operasies wat op die objek uitgevoer word nie. Die beskermingsmeganisme van die model word voorsien op die objekvlak, wat gebruik van die model op multi-gebruiker toepassings moontlik maak.

BASISDATAMODEL

'n Model vir magtiging moet ook ontwerp word om konsekwent te wees met die basis datamodel wat ondersteun moet word in die databasisstelsel, bv. as die databasis wat ondersteun moet word 'n objekgeoriënteerde databasis is moet die datamodel ook objekgeoriënteerde konsepte ondersteun[Rab91].

In die stel van die eienskappe van die model moet die basisdatamodel waarop die stelsel gebou word, uiteengesit word. Dit kan bv. 'n Semantiese assosiatiewe model wees. Die datamodel wat as basis voorgestel word, kan sekere eienskappe bevat wat die sekerheidsmodel kan beïnvloed. 'n Semantiese assosiatiewe model is byvoorbeeld saamgestel uit saamgestelde objekte wat bestaan uit 'n versameling feite en 'n

versameling van relevante reëls. Die saamgestelde objekte mag die werking van die sekerheidstelsel bemoeilik en daarom moet die gevolge wat dit op die sekerheidsmodel mag hê, in ag geneem word.

Die faktore wat 'n invloed mag uitoefen op die sekerheidstelsel word ook voorgestel. Daar sal byvoorbeeld gesê word hoe die objekte verbind word, soos byvoorbeeld vir die semantiese assosiatiewe model sal objekte met kennisreëls of databasisfeite verbind word. Al die kennismanipulasiebewerkings van dié semantiese assosiatiewe model kan gebruik word om die toegangsreëls uit te druk. Sommige van die reëls kan integriteit of sekerheidsreëls wees, d.w.s. hulle kan die basis meganisme wees om integriteit of sekerheid af te dwing[Lar90].

KARAKTERTREKKE

'n Ander gedeelte van die uiteensetting van 'n model is die beskrywing van die model se karaktertrekke. 'n Model moet soms sekere goeie eienskappe prys gee om die doel waarvoor die model gebou word na te streef, maar ook om die beleid wat die model moet ondersteun te akkommodeer. Dit is daarom raadsaam om vir die gebruiker die karaktertrekke van die model uiteen te sit. Voorbeelde van karaktertrekke sluit die volgende in:

Buigbaarheid
Die ondersteuning van 'n klassifikasiestruktuur,
Hoër- of Laervlakbeskerming
Beskerming met behulp van vermoëns (sleutels)
Beskerming vir gebruikersrolle

VOORBEELDE VAN UITEENSETTINGS

'n Paar modelle sal nou voorgestel word met hul doel en eienskappe, om as voorbeeld te dien van die uiteensetting van 'n model. Ons kyk byvoorbeeld na die DISCO[Oli91],SODA[Kee89], DAMOKLES[Dit89], en die PRIVATE TOEGANGSKANAAL[Dol93] model.

Die uiteensetting van **SODA[Kee89]** is soos volg :

DOEL

Die doel van SODA is om 'n buigbare sekerheidsmodel daar te stel, wat 'n buigbare dataklassifikasiemodel (gebaseer op oorerwing) ondersteun.

BASIS

Die basis van SODA is 'n objekgeoriënteerde databasis.

KARAKTERTREKKE

Die gebruik van 'n klassifikasiebeleid maak toelating vir 'n gladde oorgang tussen rigiede klassifikasiereëls en onbeperkte poli-instansiering.

Daar word gebruik gemaak van die stelselsekerheidsbeampte vir klassifikasie definiëringsdoeleindes.

Die sekerheidsmodel hanteer die datamodel sowel as die berekeningsmodel van objekgeoriënteerde stelsels, en bied meer buigbaarheid.

Die model verruil 'n toename in kompleksiteit vir 'n meer buigbare model.

Die doel in hierdie model was om 'n meer buigbare sekerheidsmodel daar te stel wat toelaat vir komplekse aspekte soos poli-instansiëring, ens. 'n Ander belangrike aspek van die model is, dat omdat alles in die objekgeoriënteerde modelobjekte is, en die objek die beskermde entiteit is, daar 'n goeie versekering bestaan dat geen gedeelte van die stelsel onbeskermd is nie.

Die uiteensetting van DISCO[Oli91] is soos volg:

DOEL

Die doel van die model is om die implikasies wat kan vorm met die oorplasing van 'n vermoë (sleutel tot entiteit) aan 'n ander subjek, te bestudeer, en om die beperkings wat van toepassing is op só 'n oorplasing, te identifiseer.

BASIS VAN MODEL

Diskresionêre sekerheid word toegepas op objekgeoriënteerde databasisse,

KARAKTERTREKKE

Entiteite in die databasis word beskerm deur vermoëns. (Dit is 'n onvervalsbare teken wat die verwerker in staat stel om 'n verwante entiteit op 'n manier te gebruik, of toegang daartoe te verkry.)

'n Subjek wat 'n vermoë besit, word gemagtig om toegang tot die ooreenstemmende entiteit te verkry.

'n Subjek mag 'n vermoë aangee of oorplaas na 'n ander subjek om so aan die ander subjek toegang tot die beskermde entiteit te verskaf.

Hierdie oorplasing van die magtiging word gedoen op die diskresie van die eerste subjek.

Wegneem van vermoëns word voorsien in die model maar die probleme wat kan voorkom met die wegneemproses word in meer besonderhede hanteer,

Die doel van hierdie model is dus van 'n wetenskaplike aard en kan gebruik word in die uitbou van bestaande modelle, of om tekortkominge in bestaande modelle te oorkom.

Die uiteensetting van DAMOKLES[Dit89] is soos volg:

DOEL

DAMOKLES beskerm gestruktureerde objekte in geheel sowel as die kompartemente of gedeeltes van die objek afsonderlik.

Die veranderde struktuur van saamgestelde objekte word beheer deur magtiging met komplekse regte voor te stel.

As komponente van 'n objek verskillende eienaars het wat verskillende besluite neem oor die toekenning van voorregte, is daar geen manier om uniforme toegangsregte vir die hele objek te gee nie. Die gebruik van 'n databasissleutel is die oplossing tot hierdie probleem.

Slegs die eienaar van 'n objek kan regte toeken aan ander, d.w.s. hierdie model is ook gebaseer op diskresionêre sekerheidsbeheer.

BASIS VAN MODEL

DAMOKLES is gebaseer op struktureel-objekgeoriënteerde databasisse. Die toegangsbeheer komponent van DAMOKLES is dan ook verteenwoordigend van hierdie tipe objekgeoriënteerde databasisse.

KARAKTERTREKKE

Die doel van die model is om 'n diskresionêre sekerheidsmodel daar te stel wat gebaseer is op struktureel-objekgeoriënteerde databasisse.

Daar bestaan nog 'n opsie vir sekerheidsmodelle wat min modelle nog gebruik, dit is die gebruik van gebruikersrol-gebaseerde sekerheid. Dit is 'n tegniek vir die karakterisering van databasissekerheid waar die verantwoordelikheid van die individu in ag geneem word, as die sekerheidsvereistes van die toepassing gedefiniëer word.

T C Ting[Tin92] et al, verklaar dat die gebruik van gebruikersrol gebaseerde sekerheid 'n goeie tuig bied vir die karakterisering van regte en voorregte van individue wat toegang benodig tot die komplekse toepassings.

Die model vir Rol-gebaseerde toegangsbeheer voorsien 'n magtigingslys vir die voorstelling van gebruikersrolle van die individue wat toegang tot die toepassing benodig, en dit voorsien ook 'n versameling analisetegnieke wat 'n raamwerk vorm vir die ontwerper om te verstaan, en ook om die sekerheid spesifikasie te evalueer en reg te maak[Tin92].

Die definisie van gebruikersrol-gebaseerde sekerheid is die eerste stap vir die ontwerper om die toegangs- en magtigingsvoorregte vir hul toepassings te identifiseer en daar te stel. Om die ontwerper in die proses by te staan, word 'n raamwerk vir identifisering van die gebruikersrolle voorsien en dit word die gebruikersroldefinisie-hiërargie genoem. As die Gebruikersroldefinisiehiërargie eers saamgestel is, moet die ontwerper die magtigingvoorregte verskaf vir die hiërargie deur die toekenning van metodes aan die hiërargienodus. As die metodetoekenning gedeelte voltooi is, is dit moontlik om outomaties die eienskappe van die metodes wat toegeken en geërf word vir die gebruikersrolle, te sintetiseer.

Uit die bostaande voorbeelde is dit duidelik dat daar met alle modelle 'n doel is waarom die model gebou is. Heelwat van die modelle word gebou om sekere tekortkominge te oorbrug. Indien die nuwe model nou opgestel word kan daar gekyk word na die probleme wat hierdie modelle oorbrug het, om sodoende nie probleme in die nuwe model wat opgestel word in te bou nie. Daar kan ook gekyk word na die voordele en uitstaanpunte van hierdie modelle om dan te gebruik in die bou van die nuwe model. Let egter daarop dat die meeste modelle verskeie beleidsrigtings volg, hetsy die beleidsrigting van die organisasie of algemene sekerheidbeleidsrigtings. Hierdie beleidsrigtings speel 'n belangrike rol in die klassifisering van die sekerheidsmodel en sal vervolgens bespreek word.

5.3. BELEIDSRIGTINGS

Die uiteensetting van die voorgestelde model is nou voltooi, maar die voorgestelde model moet aan sekere sekerheidsmaatreëls of beleidsrigtings gehoor gee. Dié maatreëls of beleidsrigtings word óf deur die maatskappy gespesifiseer as afdwingbaar of dit word ingebou in die ontwerp van die model om aan die model 'n graad van sekerheid te verskaf. Sekerheidsmodelle word byvoorbeeld ingedeel in verskeie grade van veiligheid deur die TCSEC klassifisering. Die grade van veiligheid strek van 'n C3 sekerheidsvlak tot 'n A1 sekerheidsvlak[Pfl89]. (Vir meer besonderhede sien hoofstuk een)

Die beleidsrigtings wat gewoonlik in 'n sekerheidsmodel geïmplementeer kan word, sluit algemene databasissekerheidsbeleide, objekgeoriënteerde beleidrigtinge en administratiewe beleidsrigtings in. Die keuse van beleidsrigtings vir sekerheid is belangrik omdat dit die buigbaarheid, bruikbaarheid en werksverrigting van die stelsel kan beïnvloed[Lar90]. Die drie kategorieë van beleidsrigtings sal vervolgens bespreek word.

5.3.1. ALGEMENE DATABASISSTELSELBELEIDSRIGTINGS

'n Samehangende versameling beleidsrigtings word benodig as 'n riglyn vir die ontwerp en gebruik van 'n databasissekerheidstelsel[Lar90]. Die bestaande databasis beleidsrigtings word afgedwing in die volgende tipes gebruike:

(a) Die gebruik van 'n Oop vs 'n Geslote stelsels
(b) Eienaarskap vs. Administrasie van entiteite
(c) Diskresionêre vs. Multivlaksekerheid.
(d) Grein.

5.3.1.1. OOP VS GESLOTE STELSELS

'n Fundamentele keuse wat gemaak moet word in die ontwerp van 'n databasisstelsel, is tussen 'n oop of 'n geslote stelsel. In 'n oop stelsel word toegang verskaf tot alle entiteite (beskermbare elemente) tensy toegang verbied word, terwyl toegang tot alle entiteite geweier word in 'n geslote stelsel, behalwe indien toegang toegeken word. Goeie sekerheid vereis geslote stelsels, terwyl buigbaarheid 'n aanduiding van 'n oop stelsel is. In die algemeen word 'n geslote stelsel gebruik as 'n hoë graad van sekerheid 'n belangrike doelwit is[Lar90].

Die magtigingstelsel wat gebruik word in die "SEKERHEIDSBELEID VIR GEïNTEGREERDE PROJEKONDERSTEUNINGSOMGEWINGS"[Der90] maak gebruik van 'n geslote stelsel. Dit word geïmplementeer deur aan die gebruiker geen toegang te verskaf aan enige domein in die databasis, voordat magtigingsreëls spesifiek vir die gebruiker gespesifiseer word nie. Dit wil sê dat toegang slegs verkry word nadat dit toegeken is.

5.3.1.2. EIENAARSKAP VS ADMINISTRASIE

Die ***Eienaarskapbeleid*** kan omskryf word as die beleid waar die gebruikers die eienaar van data is wat hul geskep het en waar dié eienaars hul eie data administreer. In die ***beleid van administrasie***[Lar90] word die omgewing gesien as die eienaar van die inligting en die gebruikers ontvang toegang tot data wat aan hulle gegee word om hul funksies daarop te kan uitvoer, terwyl spesiale gebruikers die gebruik van inligting administreer. Die feit dat 'n Databasisbeheerstelsel of 'n kennisbasisbeheerstelsel gebruik word om die omgewing te ondersteun, maak administrasie in sommige gevalle 'n meer logiese keuse as beleid vir 'n magtigingsmodel.

In die DAMOKLES model[Dit89] verkry 'n gebruiker die EIENAAR-eienskap vir die onderliggende beskermde objekte of p-objekte. Eienaarskap word hier verkry indien hy 'n databasis, objek of verwantskap skep, oftewel indien hy 'n entiteit skep. Dié eienaar besit alle regte vir die geskepte elemente wat hy mag toeken of

wegneem aan of van ander subjekte. 'n Eienaar is 'n gebruiker U en nie 'n subjek paar (U[Gebruiker],P[Program]) nie, d.w.s. as hy regte vir sy objek wil uitdeel moet hy aan subjekte (U[Gebruiker],P[Program]) die regte toeken, d.w.s. slegs die eienaar kan regte vir p-objekte toeken en wegneem. Die eienaar-eienskap mag oorgeplaas word aan ander gebruikers of gebruikersgroepe, maar daar is te eniger tyd presies net een eienaar vir elke p-objek. In die geval van gebruikersgroepe sal die administrateur die toeken en wegneemoperasies uitvoer. 'n Voordeel hiervan is dat die dinamiese en gedesentraliseerde magtiging bevorder word[Lin89].

5.3.1.3. DISKRESIONêRE VS MULTIVLAKSEKERHEID

Die keuse tussen die gebruik van diskresionêre en multivlaksekerheid is nog 'n belangrike beleidsbesluit wat geneem moet word.

In multivlak, ook bekend as verpligte sekerheid[Lar90], word objekte geklassifiseer in sekerheidsvlakke en beheer word gedefinieer oor die vloei van inligting tussen vlakke[Lar90].

SODA[Kee89] maak gebruik van 'n multivlaksekerheid objekgeoriënteerde databasis met die volgende eienskappe: (a) Inligtingbevattings- en sekerheidsmerkingsintegriteit word afgedwing, (b) die model omvat 'n berekeningsmodel, sowel as datatoegang en -klassifikasie, en (c) dit maak voorsiening dat die klassifikasievlak van 'n proses aangepas word, gebaseer op sy klaringsvlak wat moontlik is (die data waartoe dit toegang het) en laastens word klassifikasiebeperkings toegepas op die data wat geskep word. Dié klassifikasiemetode laat toe dat die klassifikasierigiedheid gekontroleer word op 'n klas-vir-klas-basis. Dit maak ook voorsiening vir 'n wye verskeidenheid merkingsvereistes[Kee89].

Multivlak-veilige rekenaars beskerm objekte wat geklassifiseer word op meer as een vlak en laat deling tussen gebruikers van verskillende klaringsvlakke toe. In so 'n multivlaksekerheidstelsel word objekte gemerk met hul sensitiwiteitsvlakke[Kee89] en subjekte word geassosieer met klaringvlakke en die

kombinasie van die sensitiwiteit- en klassifikasievlakke word gebruik om toegangregte te definieer . 'n Multivlak-veilige rekenaar arbitreer alle toegang van objekte deur subjekte. Die arbitrasie word gewoonlik uitgevoer deur 'n verwysingsmonitor volgens 'n sekerheidsbeleid[Kee89].

Die gebruik van die Multivlaksekerheidsbeleid impliseer dus dat daar aan alle subjekte en entiteite in die stelsel óf 'n klassifikasievlak of 'n sensitiwiteitsvlak toegeken moet word. Toegang tot entiteite word dan gebaseer op hierdie sensitiwiteits- en klassifikasievlakke. Alhoewel dit dan tyd vereis om hierdie klassifiserings te doen, word multivlaksekerheid gesien as die veiliger opsie van sekerheidsmeganismes en mag dit raadsaam wees om hierdie keuse op te neem. Indien die veiligheid van die stelsel egter nie só belangrik is nie kan daar gebruik gemaak word van diskresionêre sekerheid.

In diskresionêre sekerheid word die manier waarop individuele subjekte (gewoonlik die individu wat eienaarskap bevat) spesifieke objekte manipuleer, gespesifiseer. Eksplisiete magtiging ken toegang toe deur eksplisiet gebergde reëls en implisiete magtiging [Lar90] laat toe dat die stelsel effektiewe magtiging aflei vanuit magtiging wat eksplisiet gestoor word in die stelsel. Diskresionêre sekerheid kan gekombineer word met multivlaksekerheid om 'n hoë sekerheidstoepassing te verkry[Lar90].

'n Aspek van diskresionêre en multivlaksekerheid wat dus in ag geneem moet word is die area waar die twee tipes gekombineer word. In hierdie area moet daar reëls of 'n beleid saamgestel word wat sal uitwys wanneer diskresionêre sekerheid, die multivlaksekerheid oorheers en omgekeerd[Der90].

5.3.1.4. GREIN.

Die grein van die data-objekte in die toegangsreëls is 'n ander beleidsbesluit wat geneem moet word. Daar bestaan drie tipes kontroles naamlik naam-afhankike, inhoud-afhanklike, en konteks-afhanklike beheer, wat afgedwing kan word.

In ***naam-afhanklike toegangsbeheer*** kan toegang aan een gebruiker toegeken word tot die algehele klas terwyl daar aan 'n ander gebruiker net toegang toegeken word tot spesifieke attribute in die klas.

In ***inhoud-afhanklike toegang***, resulteer toegang in 'n fyner mate van beheer deur toegangreëls te spesifiseer wat verwys na die inhoud van die data item voorkomste (sowel as na hul name). Dit word ook predikaat-gebaseerde toegangsbeheer genoem.

Die beleid van ***konteks-afhanklike toegangsbeheer*** verwys na kombinasies van items en die konteks waarin hulle voorkom en beperk die velde wat saam gebruik kan word.

5.3.1.5. INTEGRITEIT SEKERHEIDSBELEID

Die A1-veilige databasis beheerstelsel dwing die streng skryfeienskap of **integriteitsekerheidsbeleid** van die BIBA-integriteitsmodel soos volg af[29]:

(a) As 'n program voorkoms ekstern tot die betroubare rekenbasis 'n ry byvoeg in 'n databasistabel, moet die integriteitsvlak van die program, die integriteitsvlak van die ry domineer,

(b)'n Program voorkoms ekstern tot die betroubare rekenbasis kan 'n ry verwyder of wysig slegs as sy integriteitsvlak dié van die ry domineer.

5.3.2. BELEID VIR OBJEKGEORIËNTEERDE DATABASISSTELSELS

Die tweede afdeling van beleidsrigtings wat ingebou kan word in 'n sekerheidsmodel om dit veiliger of doeltreffender te maak, is die objekgeoriënteerde databasisbeleidsrigtings.

Daar bestaan drie tipes objekgeoriënteerde databasisstelsels en elk van hulle beïnvloed die sekerheidsbeleid op 'n ander wyse, afhangende van die eienskappe van die stelsel. Die drie tipes is strukturele objekgeoriënteerde databasisstelsels, gedragsobjekgeoriënteerde databasisstelsels en ten volle objekgeoriënteerde databasisstelsels.

Stukturele objekgeoriënteerde databasisstelsels voorsien meganismes (strukture en generiese operatore) wat die saamgestelde databasis objekte (d.w.s. objekte wat 'n resultaat is van die samesmelting van ander objekte in 'n willekeurige manier) hanteer.

Gedragsobjekgeoriënteerde databasisstelsels laat toe dat gebruikers willekeurige tipe-spesifieke operatore en ook nuwe objektipes definieer. Gedrags-objekoriëntasie sluit tipies die enkapsulasie van die onderliggende waarde voorstellings struktuur in, en is dus baie geskik vir vermoë-gebaseerde toegangsbeheer tegnieke[Lin89]. DAMOKLES[Lin89] maak gebruik van strukturele objekgeoriënteerde databasisse.

Ten volle objekgeoriënteerde databasisstelsels kombineer die eienskappe van strukturele en gedrags-objekgeoriënteerde databasisstelsels.

Aspekte wat beleidsrigtings of beleidsbesluite in die objekgeoriënteerde omgewing sluit die volgende in:

(a) oorerwing,

(b) sigbaarheid van onder,

(c) sigbaarheid van bo,

(d) negatiewe magtiging,

(e) implisiet spesifiek

(f) predikate en

(g) die dataklassifikasiebeleid.

Hierdie aspekte sal vervolgens verder uitgelig word.

5.3.2.1. OORERWING

Oorerwing is een van die mees beduidende konsepte in objekgeoriënteerde programmering. Die effek wat dit het op sekerheid, is dat toegang tot 'n versameling klas attribute dieselfde tipe toegang tot dieselfde attribute in sy subklasse kan impliseer.

Maria M Larrondo-Petrie, et al.[Lar90] se beleid I is 'n oorerwingsbeleid en sê dat 'n gebruiker wat toegang tot 'n klas het toegelaat sal word om dieselfde tipe toegang te verkry tot die attribute in die ooreenstemmende subklasse wat geërf was vanaf die klas.

5.3.2.2. SIGBAARHEID VAN ONDER

Sigbaarheid van bo en onder is in 'n mate ook 'n gevolg van oorerwing, maar beïnvloed wel die ontwerp van die sekerheidsmodel en daarom moet dit in die sekerheidsmodelbeleid ingesluit word of weggelaat word.

Sigbaarheid van onder is waar toegang tot 'n subklas, toegang tot attribuut waardes van 'n superklas, wat ooreenstem met die subklas, impliseer. Die beleid stem ooreen met Larrondo-Petrie, et al.[Lar90] se Beleid II.

Beleid II[Lar90] sê dat toegang tot 'n volledige klas, toegang tot die attribute gedefinieer in daardie klas impliseer, sowel as toegang tot die attribute geërf vanaf

'n hoër klas. As daar meer as een voorouer is, word toegang verkry tot die vereniging van die geërfde attribute.

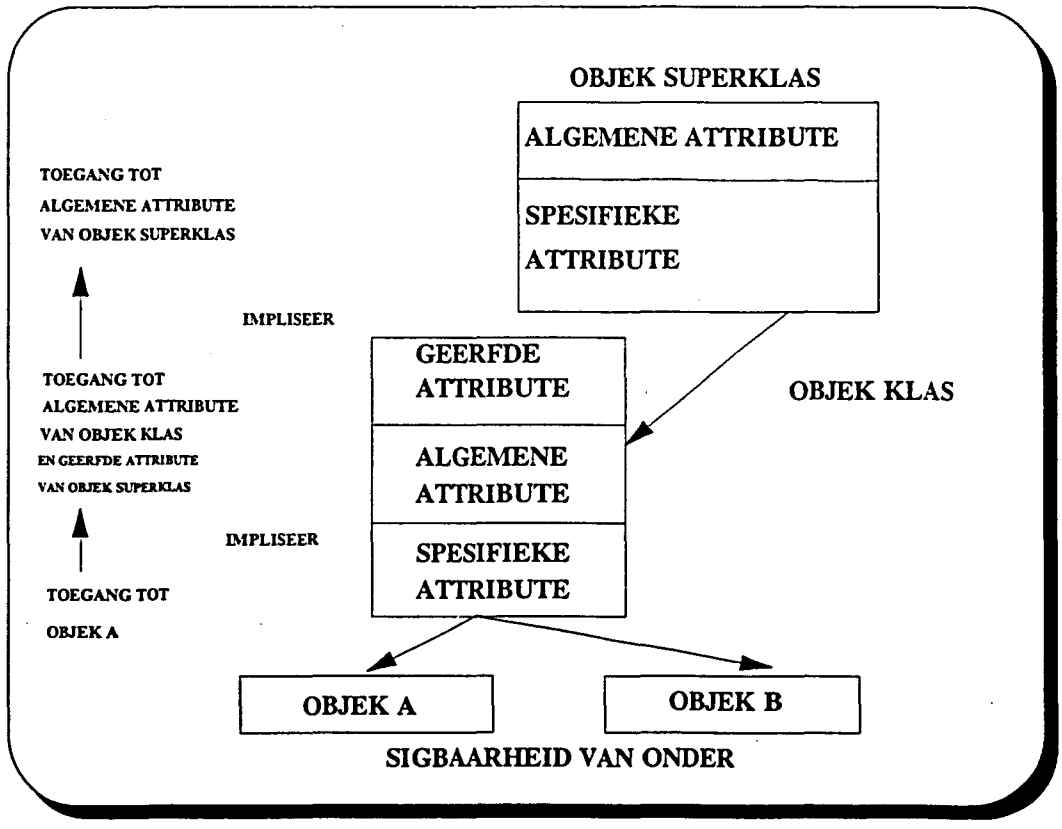


Fig 5.2. Sigbaarheid van onder

5.3.2.3. SIGBAARHEID VAN BO

Daar bestaan twee beleide wat sigbaarheid van bo kan beïnvloed. Die keuse tussen die twee hang van die tipe objekgeoriënteerde stelsel wat vereis word af. Daar bestaan drie verskillende tipes objekgeoriënteerde stelsels en hulle is toepassing afhanklik[Lar90].

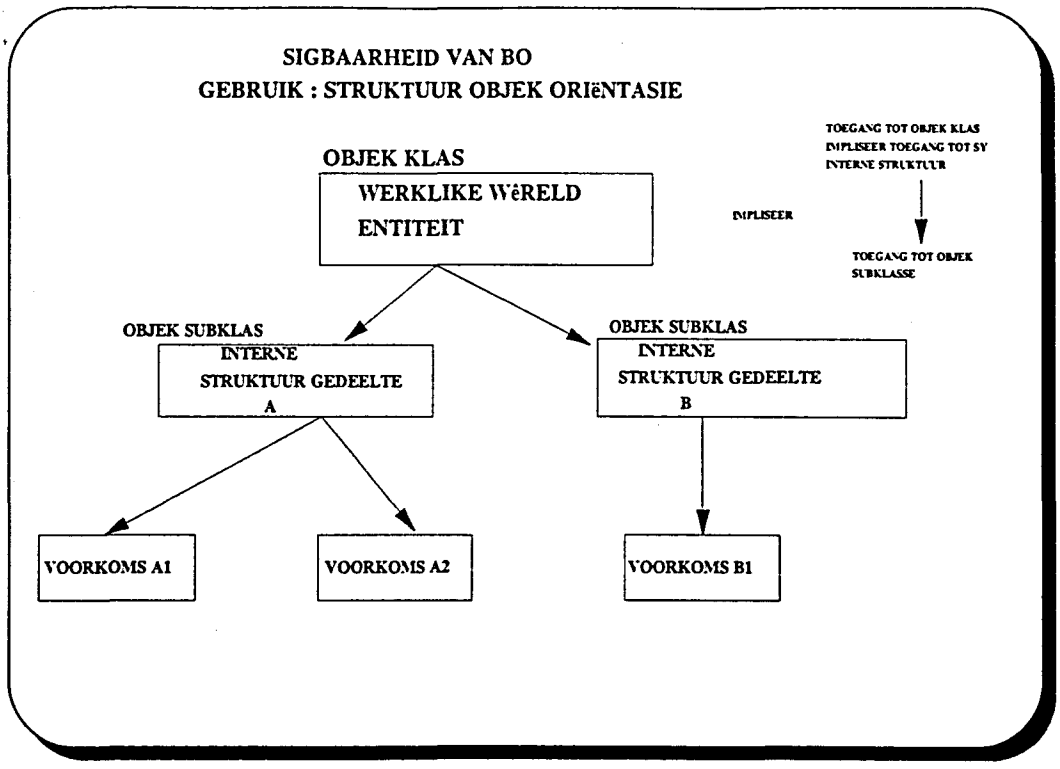


fIG 5.3. Sigbaarheid van bo

Struktuur-objekgeoriënteerdheid verskaf die vermoë om objekte met komplekse interne strukture te definieer en te manipuleer. Die objekte word gesien as saamgestelde objekte wat georganiseer word in objekhiërargieë. Die wortel objek is 'n "werklike wêrelde" objek, terwyl die ander sy interne struktuur beskryf. Die beleid vereis vir struktuur-objekgeoriënteerde databasisse is dat die magtiging om 'n objek te gebruik, toegang tot al die ander objekte wat sy interne struktuur beskryf impliseer. Dit word die "STERK VERBINDINGS" beleid genoem. Die beleid sal vereis word vir CAD-/CAMdatabasisse om dit in staat te stel om verkleining en vergroting van objekte te doen. 'n Attribuut gedefinieer vir 'n subklas IS toeganklik (of sigbaar) deur enige van sy superklasse te gebruik.

Larrondo-Petrie, et al.[Lar90] se beleid III is konsekwent met 'n gedrags-objekgeoriënteerde stelsel, wat toelating maak dat tipe-spesifieke operatore gedefinieer word vir objekte, met enkapsulering en oorerwing van operatore in 'n oorerwingshiërargie. In gedrags-objekgeoriënteerde databasisse, word objekklasse nie sterk verbind nie, en daarom impliseer toegang tot 'n klas nie toegang tot al die attribute in sy subklasse nie. Beleid III[Lar90] word die

"SWAK OF LOS VERBINDINGS"-beleid genoem, en spesifiseer dat 'n attribuut gedefinieer vir 'n subklas NIE toeganklik is deur enige van sy superklasse te gebruik nie. Dit is die teenoorgestelde van die bogenoemde en dus moet die tipe sigbaarheid van bo duidelik uitgewys word in die sekerheidsbeleid.

Die derde siening is 'n TEN VOLLE objekgeoriënteerde stelsel, dit kombineer aspekte van beide die struktuur-objekgeoriënteerde en gedrags-objekgeoriënteerde stelsels om sekerheidsvereistes te definieer. Die beleid wat benodig word onder sulke stelsels is toepassingsafhanklik.

5.3.2.4. NEGATIEWE MAGTIGING

Negatiewe magtiging is 'n reël waar toegang tot 'n entiteit eksplisiet geweier word [Rab91]. 'n Gebruiker kan in die algemeen faal om toegang te verkry tot die gebruik van 'n entiteit te verkry onder twee omstandigdhede: (1) as die gebruiker geen magtiging het vir die entiteit nie, of (2) as die gebruiker 'n negatiewe magtiging op die entiteit het.

Positiewe magtiging is wanneer daar aanvaar word dat toegang tot alle entiteit 'null' is, tensy die toegang gespesifiseer is. D.w.s. waar daar nie toegang gespesifiseer is nie, word 'n 'null' of geen-waarde-toegang aanvaar, tensy daar 'n spesifieke negatiewe toegang gespesifiseer word met negatiewe magtiging.

In die algemeen dek die volgende vrae inligting wat in die beleidsrigting rakende negatiewe magtiging gespesifiseer moet word, nl.:

(a) Wanneer moet negatiewe magtiging positiewe magtiging oorskryf?
(b) Wanneer moet positiewe magtiging negatiewe magtiging oorskryf?

Vir die volgende gedeelte van die bespreking word daar veronderstel dat elke omgewing verdeel kan word in drie dimensies, naamlik 'n Magtigingsdimensie, 'n

Subjekdimensie en 'n Entiteitsdimensie, waar toegang as 'n funksie op dié drie-dimensionele omgewing gespesifiseer word.

Oorerwing maak die probleme nog groter deur vier aparte kondisies in toegangsbeheer te veroorsaak wat moontlik ook gesamentlik kan bestaan :

Eksplisiete Weiering

Dieselfde as negatiewe magtiging

Eksplisiete Magtiging

'n Eksplisiete magtiging impliseer magtiging langs enige kombinasie van die drie dimensies in magtigingsdefinisies, naamlik: die subjek, die magtigingstipe en die magtigingsentiteit[Rab91]. 'n Voorbeeld hiervan is 'n bywerkingstoegang vir 'n lid van die "groep gebruikers" op 'n "groep objek" kan 'n bywerking vir enige lid van die groep gebruikers op enige lid van die groep objekte impliseer of dit kan 'n lees vir enige lid van die gebruikers op die groep objekte vir sy lede impliseer[Rab91].

Implisiete Weiering

Die weiering van toegang tot 'n entiteit a.g.v. die weiering van toegang tot 'n entiteit hoër op in die hiërargie.

Implisiete Magtiging

'n Magtiging van 'n sekere tipe gedefinieer vir gebruikers op 'n sekere databasisobjek impliseer ander magtigings[Rab91].

Die konsep van implisiete magtiging maak dit onnodig om alle magtigings eksplisiet te stoor, want die magtigingsmeganisme kan die magtigings bereken vanuit 'n minimale versameling van eksplisiet gestoorde magtigings. Die oorhoofse koste van hierdie berekenings moet egter opgeweeg word teen die koste van die storing van eksplisiet gedefinieerde reëls.

'n Beleid wat voorkeur tussen hierdie kondisies bepaal, moet gekies word, bv. [Lar90] kies dat voorkeur gegee word oor Positiewe magtiging, d.w.s (Weiering > Magtiging), en eksplisiete magtiging het voorkeur oor Implisiete Magtiging, d.w.s. (Eksplisiet > Implisiet).

IMPLISIETE SPESIFIEKHEID

In [Lar90] word die sterkte van 'n reël verwant gestel aan die ligging van die objek in die hiërargie. Vrae wat bv. gevra word is (a) Hoe affekteer die ligging van die objek in die hiërargie die sterkte van die reëls? (b) Moet reëls wat verwys na objekte wat nader aan die klas is, dominerend wees oor die wat verder vanaf die superklasse is, of moet die teenoorgestelde waar wees?. Hierdie aspek word implisiete spesifiekheid genoem[Lar90].

PREDIKATE

Predikate of eienskappe vir 'n toegangsreël kan ekplisiet of implisiet wees, dit kan ook meer of minder beperkend wees[Rab91]. Dit is duidelik dat daar 'n beleid moet wees wat waarborg dat toegang tot die attribute van 'n klas, die predikate van daardie klas moet bevredig, sodanig dat magtiging aan die gebruiker gegee word om toegang te verkry. Die vraag wat hier gevra word, is egter hoe die predikate vir die klas bepaal word.

Moet 'n eksplisiete reël wat 'n predikaat bevat alle geërfde reëls oorskryf wat predikate bevat, of moet die beleid wees om die vereniging van eksplisiete en implisiete predikate te neem, of moet die beleid wees om die deursnede van die predikate te neem?.

5.3.2.5 DATAKLASSIFIKASIEBELEID

'n Klassifikasiebeperking bestaan uit twee dele, nl. die eerste deel spesifiseer die tipe merking, objek of voorkomsveranderlike, en die tweede deel bestaan uit 'n sensitiwiteitsvlak-reeks vir elke gemerkte objek[Kee89]. In die geval van objekmerking is daar een reeks vir die objek self. Vir voorkomsveranderlike merking is daar een reeks vir elke voorkomsveranderlike. Die reeks spesifiseer toelaatbare limiete op die sensitiwiteitsvlak van die beskermde objek.

Data klassifikasie word gebaseer op die oorerwingstralie van die stelsel[Kee89]. Klassifikasie word bepaal op twee maniere, nl. tipe en spesialisasie.
(1) Die klassifikasie beperking vir 'n objek word verkry vanaf sy klas.
(2) Die klassifikasie beperkingsreeks van 'n nuwe klas word geërf vanaf sy supertipe.

Die oorerwingstralie voorsien 'n natuurlike manier vir kategorisering van objekte in semanties betekenisvolle groepe. Elke klas word vanaf (1) 'n klassifikasie beperking toegeken wat van toepassing is op al sy voorkomste.

Die klassifikasiereeks van 'n subklas word geërf vanaf sy superklas(2). Die merkingstipe en reeks vir ooreenstemmende merke is dieselfde. In die geval van veranderlike merking mag die subtipe addisionele veranderlikes verklaar. Hierdie veranderlikes is onbeperk, d.w.s. hulle het 'n reeks van [stelsel laag, stelsel hoog]. Dit laat die gebruiker toe om nuwe klasse te skep. Hulle word egter nie toegelaat in beperking te verander wat deur die stelselsekerheidsbeampte geskep is nie[Kee89].

Die klassifikasiemeganisme kan gebruik word om assosiasie tussen objekte te versteek. Klassifikasie kan gebaseer word op die waarde van 'n voorkoms veranderlike. Dit vereis die skepping van 'n subtipe vir elke klassifikasie groep.

Beperkinge wat gestel word deur die klassifikasiebeperking word afgedwing deur die TCB. AS 'n sensitiwiteitsvlak vir 'n nuwe objek nie gevind kan word wat beide die klassifikasie beperking bevredig en inligting bevatting[Kee89] in stand hou nie, sal die nuwe objek nie geskep word nie.

Sensitiwiteitsvlakreekse voorsien 'n eenvoudige manier om data te klassifiseer. Hierdie tegniek voorsien 'n eenvoudige afdwinging wat uitgevoer kan word in die TCB. Die taak van toekenning van beperking en hul verifikasie word gedoen deur 'n aparte betroubare applikasie, die sensitiwiteitsmerker.

5.3.3. BELEID MET BETREKKING OP ADMINISTRATIEWE REGTE

Die vermoë om sekerheid en skemaregte te delegeer oor 'n deelversameling van die databasis aan 'n ander databasisadministreerder bring nuwe aspekte en beleide wat beskou moet word. Sekerheid- en skemaregte moet apart beskou word om meer buigbaarheid te verskaf vir die delegering en vernietiging van administratiewe regte[Lar90]. Sekerheidsregte mag toegeken word oor 'n deelversameling van die databasis sonder dat die regte om skemaverandering te maak oor daardie deelversameling van die databasis besit moet word. Sekerheidsregte moet egter ook gedelegeer om sodoende integriteit en konsekwentheid in stand te kan hou as skemaregte toegeken word, en ook om sodoende die nuwe administreerders toe te laat om toegangsreëls te definieer vir die veranderde konteks. Met die delegering behou die delegerende administreerder die regte om die regte wat gedelegeer is weg te neem.

'n Beleidsbesluit word benodig om te bepaal wat moet gebeur met die maatskappy- en modelreëls na delegering.

Die bogenoemde beleidsrigtings kan almal gebruik word in die samestelling van 'n beleid vir die model, of dit kan gebruik word in die ontwerp van die nuwe model.

Hoe dit ookal sy, afhangende van die tipe ontwerp wat die model sal gebruik sal verskillende kombinasie van die bogenoemde beleidsrigtings die ontwerp komplementeer of afbreek, wees dus versigtig in die keuse van beleidsrigtings.
Die volgende bespreking wat gedoen word is die uiteensetting van die model, d.w.s. uit watter komponente bestaan 'n tipiese model en hoekom word juis daardie komponente gebruik?

5.4. UITLEG VAN DIE MODEL

Verskeie besluite moet geneem word met die uitleg van die model, bv. wat om te beskerm, hoe om dit te beskerm, en hoe om skadelike optrede teenoor die omgewing in die model te voorkom. Ons het besluit om die riglyne van Klaus Dittrich et al[Lin89] te gebruik as voorbeeld vir besluite wat geneem moet word in die bou van die model. Die riglyne wat hulle voorstel vir die bou van die model is die volgende :

1.Stel die passiewe elemente of objekte van die stelsel voor, dit is die elemente waartoe toegang benodig word deur die aktiewe elemente.

2.Stel die aktiewe elemente of subjekte van die stelsel voor, in DAMOKLES[Lin89] se geval is die aktiewe elemente die wat toegang tot ander elemente benodig.

3.Nadat die passiewe en aktiewe elemente nou voorgestel is, moet uitgeklaar word, watter tipe aksies deur die aktiewe elemente uitgevoer kan word op die passiewe elemente, bv. lees, skryf, bywerk, skrap, ens. Dit is indien dit nodig is om dit eksplisiet te spesifiseer.

4.Die omgewing is nou uitgeklaar, en nou moet die meganisme wat toegangsbeheer hanteer, gemodelleer en omskryf word. Let veral hier op die beleidsrigtings wat van toepassing is op hierdie model.

5.In alle modelle moet duidelikheid verkry word oor uitsonderlike omstandigdhede, dus sal die volgende stap wees om reëls of 'n meganisme te verskaf wat sal verduidelik wat moet gebruik as daar veranderinge in die omgewing plaasvind. In hierdie afdeling kan daar ook melding gemaak word van die meganisme wat gebruik word vir die verhoeding van verbode toegange en veranderinge in gemagtigingde inligting.

Die res van die afdeling bespreek nou die bogenoemde vyf stappe duidelik met die aantoon van voorbeelde en gebruike in bestaande modelle.

5.4.1. PASSIEWE ELEMENTE VAN DIE MODEL.

(Wat word beskerm?)

Met die bou van die model moet ons besluit wat beskerm gaan word. Dié besluit word beïnvloed deur verskillende faktore, bv. die beskermingsvereistes van die maatskappy of die beleidsrigting van die model of ook die mate waarin sekerheid benodig word. Verskillende modelle gebruik verskillende entiteite as beskermde entiteite vir verskillende redes. Ons sal nou elk van hulle beskou en ook die faktore wat 'n rol gespeel het in die besluit van daardie spesifieke model.

5.4.2. ELEMENT VAN BESKERMING.

DISCO[Oli91] beskerm "ENTITEITE (of wêreldse elemente)". Dié "ENTITEITE" word voorgestel met behulp van objekte (sien Objekgeoriënteerde paradigma), want beide die entiteit se gedrag of toestand en beskrywing word in 'n objek geënkapsuleer. Die model is ook gebaseer op objekgeoriënteerde databasisse en daarom val die klem op objekte. Die entiteite as objekte word beskerm deur vermoëns wat opsigself ook objekte is, maar ons sal later aandag skenk aan hierdie unieke beskermingsobjekte. Alle "ENTITEITE" moet geskep word voordat die beskermingsmeganisme daarvoor ontwikkel of geskep kan word.

Dié denkwyse word gesien in die meeste van die modelle, daarom sal u sien dat die meeste voorbeelde wat vervolgens gegee gaan word, terugkeer na die objek as beskermde entiteit. Die gebruik van die objek as beskermde entiteit het egter sy voordele, bv. dit is alreeds geënkapsuleerd en bied 'n sekere mate van beskerming tot homself. As verduideliking kan ek bv. die volgende sê : die objek sal geen inligting oor homself net so bekend stel nie. Daar moet eers boodskappe na die objek gestuur word wat vra vir spesifieke inligting voordat dit verstrek sal word. Hierdie boodskappe kan dan onderskep en ondersoek word ter beskerming van die objek.

In SODA[Kee89] is die beskermde passiewe data of die passiewe entiteit in die model die voorkomsveranderlikes en objekte. Daar word egter gespesifiseer dat die ander keuse wat gemaak kon word, die boodskap (as beskermde entiteit) is. Die model maak gebruik van verpligte sekerheidsbeginsels en daarom word daar aan elke objek of voorkomsveranderlike 'n klassifikasie of rang toegeken om aan die *-eienskap[Pfl89] (gebruik in verpligte sekerheid) te voldoen. D.w.s. in hierdie geval moet ekstra inligting toegeken word aan die element van beskerming, ons kan dus sê ons beskerm hier 'n objek met 'n klassifikasie.

Probleme wat voorkom met hierdie tipe beskerming is dat daar nou verseker moet word dat die *-eienskap in besit moet wees van slegs objekte wat dieselfde versameling toepassing-onafhanklike eienskappe besit met verwantskappe tussen die entiteitklassifikasies. Die definiëring van integriteitsreëls vir hierdie entiteits klassifikasies moet ook geskied. Dit is moeilik om in hierdie model te bepaal watter entiteit as beskermde objekte reageer en watter as subjekte. Die klassifisering van die onderskeie objekte kan ook verander, bv. die klassifikasie van objekte met 'n onbepaalde klassifikasie word eers met looptyd bepaal, wat die sekerheidseienskappe van die model moeilik maak om te evalueer.

In DAMOKLES[Lin89] word die ENTITEIT van beskerming, p-objekte ('protected objects') genoem, en word beskryf as die kleinste eenheid van die databasis t.o.v. toegangsbeheer.

Die p-objekte word soos volg gevorm:

i. (a) 'n beskrywende deel D(die objek se eienskappe),
(b) 'n strukturele deel S(die objek se komponente met weergawes uitgesluit),
en (c) die weergawe deel V (die objek se weergawes), waar (a),(b), en (c) die objekte vorm. Die verwantskappe bestaan ook uit 'n beskrywende deel D(die verwantskap se eienskappe), en

ii. (b) die rol gedeelte R(die verwantskap se rolle). Verwantskappe en objekte vorm gesamentlik die p-objekte. Die lede S,V en R van so 'n p-objek is egter weer eens p-objekte. DAMOKLES dui ook aan dat, behalwe die bogenoemde, algehele databasisse ook p-objekte is.

Die Model vir magtiging vir die volgende generasie databasisstelsels[Rab91] stel die magtigingsobjek of eenheid van beskerming voor as óf 'n enkele objek, óf 'n groep objekte of 'n algehele databasis. In die model vir volgende generasie databasisstelsels word voorgestel dat die versameling van beskermde entiteite beskerm word met behulp van 'n versameling reëls wat gespesifiseer word in 'n drie-dimensionele omgewing waar die subjekte die een dimensie is, die beskermde entiteit die ander dimensie is en die magtigingstipe die laaste dimensie is.

Dié tipe model vereis baie tyd en beplanning vir die ontwerp van die dimensies en die reëls wat gespesifiseer moet word, maar bied 'n baie veilige sekerheidsmeganisme.

Die bogenoemde is die mees algemene keuses vir beskermde entiteite en met reg, want dit bied soveel voordele en buigbaarheid. Daar moet egter nog steeds gelet word op die beskermingsvereistes wat nodig is vir die sekerheidsmodel want dit kan beteken dat die entiteit van beskerming in kombinasie met ander dinge, soos bv. toegangsregte gebruik sal word. Daar sal vervolgens gekyk word na die aktiewe elemente van 'n sekerheidsmodel.

5.4.3. AKTIEWE ELEMENTE VAN DIE MODEL

Daar is nou al vasgestel wat die passiewe elemente van 'n model kan wees, en watter passiewe elemente vir watter tipes sekerheidstelsels gebruik moet word. Die aktiewe elemente speel egter 'n belangrike rol in die sekerheid van die passiewe elemente en sal vervolgens bespreek word.

Aktiewe elemente van 'n model is in die meeste modelle gebruikers en programme of 'n kombinasie van die twee.

Daar word egter na hulle verwys op verskillende maniere, bv. in DAMOKLES[Lin89] word die kombinasie van die gebruiker en program gesien as die subjek. Daar word verskillende vlakke met die gebruikers geassosieer, byvoorbeeld die gebruikers opsigself as een vlak, en gebruikers as groep in 'n ander vlak. Hierdie groepe kan weer in 'n hiërargie saam met ander groepe saamgegooi word om 'n supergroep te vorm, ens. DAMOKLES[Lin89] ken aan hierdie groepe, groepsadministreerders toe wat die administrasie van die groep hanteer. Die gebruikers kan ook aan meer as een groep behoort. Die Program komponent van die subjek is Programme of in objekgeoriënteerde programmeringsdefinisies boodskappe wat ook opgedeel kan word in groepe of kan apart beskou of gebruik word[Lin89]. Die subjek word in DAMOKLES geïnterpreteer as 'GEBRUIKER A TERWYL PROGRAM P GEBRUIK WORD'. DAMOKLES maak hierin 'n verfyning van magtiging, want daar word sekerheid gegee deur nie net aan gebruikers toegang tot 'n objek te gee nie, maar ook deur te sê met watter programme toegang verkry kan word.

Daar kan ook gebruik gemaak word van "gebruikersrolle" as subjekte in die stelsel, d.w.s. regte word toegeken op gebruikersrolbasis eerder as aan 'n gebruiker[Tin92]. In hierdie geval word baie tyd gespandeer aan die evaluering van verskillende gebruikersrolle. Die gebruikersrolle kan ook in verskillende hiërargieë opgedeel word en toegang kan byvoorbeeld aan rolklasse toegeken

word. Hierdie metode is veral geskik in die gebruik van objekgeoriënteerde programmering.

SODA[Kee89] soos ons weet implementeer 'n verpligte sekerheidstelsel. Dié stelsel gebruik 'n soortgelyke aktiewe entiteit as DAMOKLES maar net in 'n ander kombinasie, naamlik die subjek en boodskap gesamentlik as aktiewe entiteit. Kommunikasie in SODA geskied deur middel van boodskappe. 'n Boodskap wat aan 'n objek gestuur word saam met die gebruiker van die boodskap, verteenwoordig die subjek. 'n Boodskap is 'n versoek vanaf 'n gebruiker aan 'n objek om 'n sekere aksie uit te voer en word gemerk met die sensitiwiteitsvlak van die gebruiker gestuur.

Die boodskappe word met twee sekerheidsklassifikasievlakke gemerk, nl. (a) die klaringsvlak van die gebruiker en (b) die huidige sekerheidsklassifikasievlak van die oorsprongmetode. Dié twee vlakke reageer as 'n bo- en ondergrens van die nuwe metode-aktivering. Metode-aktiverings is die enigste aktiewe entiteite in die model. Elke metode word uitgevoer in 'n aparte konteks wat deur die aktivering beskryf word. Aktiewe entiteite word geskep as boodskappe aan objekte gestuur word. Primitiewe metodes word direk uitgevoer deur die metode-aktivering sonder om boodskappe te stuur.

D.w.s. met die spesifisering van aktiewe entiteite moet alle vlakke waarop hierdie aktiewe entiteit met die passiewe entiteit in aanraking gaan kom, duidelik gemaak word sodat, indien dit nodig is, ekstra betekenis geheg kan word aan hierdie aktiewe entiteit. Die definiëring van die aktiewe entiteit beheer in 'n groot mate die manier waarop die toegangsreëls gespesifiseer gaan word, want hoe meer besonderhede daar aan die aktiewe element geheg word, soos wie en wat hy is, hoe minder besonderhede is nodig in die identifisering van die aktiewe entiteit.

As ons nou terugkyk, het ons nou al die model uiteengesit, ons het uitgelê aan watter beleidsrigtings die model gehoor gaan gee, en ons het die elemente van die model sorgvuldig gekies. Die groot struikelblok wat nou nog in die pad lê is om

die versekering te kan gee dat alle skadelike dade (in die omgewing wat die sekerheidsmodel moet beskerm) wat gepleeg kan word, verhoed sal word. Die manier om hierdie stuikelblok te oorbrug is om na die wisselwerking tussen die aktiewe en passiewe elemente in die model te kyk en dan reëls op te stel wat slegs gemagtigde optrede sal toelaat. Die volgende afdelings sal hierdie stappe vir u duidelik na vore bring.

5.4.4. WISSELWERKING TUSSEN AKTIEWE EN PASSIEWE ELEMENTE

Die doel van die sekerheidsmodelle is om die wisselwerking tussen die aktiewe en passiewe elemente van 'n model veilig te hou. Reëls word gespesifiseer om hierdie toegange of wisselwerkinge veilig te hou.

Toegangsreëls word op twee maniere gespesifiseer, die eerste is waar reëls gevorm word uit 'n kombinasie van die aktiewe, passiewe entiteite en die tipe toegang wat gebruik word([Lin89],[Kee89],[Rab91]) en die tweede manier is waar gebruik gemaak word van 'n tipe sleutel om toegang te verleen tot die passiewe entiteite, bv. deur vermoëns te gebruik[Oli91]. In die laasgenoemde geval word daar weer reëls gespesifiseer vir die oorplasing, wegneming, kopiëring of aangee en bywerking van hierdie vermoëns.

Eerstens word die geval beskou waar die kombinasie van die aktiewe, passiewe entiteite en die toegangstipe gebruik word as magtigingsbasis.

5.4.4.1. DIE KOMBINASIE AS MAGTIGINGSBASIS.

Die kombinasie van aktiewe, passiewe entiteite en 'n toegangstipe in toegangsreëls vir die gebruik van toegangsbeheer is een manier waarop die effektiwiteit van 'n sekerheidstelsel bewys kan word. Die gebruik van toegangsreëls in die sekerheidsmodel kan ook bepaal hoe veilig die stelsel wel is.

'n Goeie voorbeeld van die kombinasie as magtigingsbasiskategorie magtigingsmetodes is die model van sekerheid vir volgende generasie databasisse[Rab91] waar die magtigingsomgewings as 'n drie dimensionele omgewing beskou word, die aktiewe entiteite is een dimensie en word gemerk met S, die passiewe entiteite vorm 'n ander dimensie, noem dit O en die laaste dimensie is die magtigingstipe domein A. Toegangsreëls word hier gespesifiseer as 'n kombinasie van SxOxA.

In die spesifisering van toegangsreëls in 'n nuwe tipe databasisstelsel bv. 'n objekgeoriënteerde databasis waar oorerwing 'n rol speel, is dit belangrik om die gevolge van die oorerwing in diepte te bestudeer en reëls op te stel om die situasies wat kan voorkom te kan oplos. Een so 'n gevolg is implisiete magtiging. 'n Voorbeeld hiervan sal nou gegee word.

In die artikel van die model vir sekerheid vir volgende generasie databasisse[Rab91] word hierdie implisiete magtiging goed voorgestel. 'n Voorbeeld hiervan is die volgende : 'n Magtiging word eksplisiet gestoor, deur bv. te stoor dat aktiewe entiteit s1 gemagtig is om toegangstipe a1 op passiewe element o1 uit te oefen. Dié spesifieke magtiging mag ander magtigings impliseer, bv. 'n gebruiker wat leesmagtiging besit tot 'n klas, moet ook leesmagtiging besit tot al die voorkomste van die klas. Dit is magtiging wat geïmpliseer word **langs die dimensie van passiewe entiteite O**. 'n Bestuurder moet toegang besit tot sy werknemer se inligting, dit is weer 'n implikasie **langs die aktiewe entiteit dimensie S** en 'n gebruiker met 'n skryf magtiging op 'n objek moet ook leesregte tot daardie selfde objek besit, wat 'n implikasie is op die **magtigingstipe dimensie A**. Hierdie model gebruik 'n funksie f met parameters (s,o,a) wat waar is as die toegang waar en korrek is. Rabbiti, et al.[Rab91] sê dat, as gevolg van die bogenoemde implikasies dit duidelik is dat reëls noodsaaklik is vir die afleiding van die waarde van $f(s_1,o_1,a_1)$ uit $f(s_2,o_2,a_2)$ en dit maak dit noodsaaklik dat alle punte vir f in SxOxA gestoor moet word, sodat daar dan ook geen konflikte meer in die waarde van f kan wees nie.

Implisiete magtiging verteenwoordig dus 'n reken-oorhoofse koste wat gebruik word om te bepaal of 'n magtiging geïmpliseer word deur 'n eksplisiet gestoorde magtiging. Hierdie koste moet geweeg word teen die bergingsmoontlikhede wat implisiete magtiging impliseer. As die vereiste vir die definiëring van magtiging op individuele magtiging nie so hoog is nie, is die geval vir implisiete magtiging ooreenstemmend laag. Vir volgende generasie databasisstelsels is die vereiste vir implisiete magtiging hoog[Rab91]. 'n Saamgestelde objek is 'n potensiële groot versameling, verwant deur 'n IS-'N-DEEL-VAN verwantskap, en 'n weergawe objek bestaan uit 'n versameling van weergawes wat verwant is deur die IS-WEERGAWE-VAN verwantskap.

Sterk Magtiging is wanneer magtigings geïmpliseer deur 'n magtigingsreël nie oorheers kan word nie. 'n Swak magtiging laat uitsonderings op hierdie reël toe.

'n Sterk magtiging waarborg dat hyself en al die magtigings geïmpliseer daardeur nie oorheers kan word nie, waar magtigings geïmpliseer deur 'n swak magtiging oorheers kan word. 'n Swak magtiging kan ook sterk uitsonderings besit.

Die model vir volgende generasie databasisse[Rab91] formaliseer die magtigingskonsepte in magtigingsreëls. Veral uitsonderlik in hierdie model is die magtigingsbasis AB wat geskep word deur deur die definiëring van alle sterk magtigings oor die drie domeine S, O en A, d.w.s AB is deelversameling van SxOxA. Die magtigings is eksplisiet sterk magtigings, dus is alle implisiete magtigings wat hieruit afgelei word ook sterk. Die model maak ook gebruik van 'n swak magtigingsbasis WAB wat alle swak magtigings oor die drie domeine S, O en A definieer, d.w.s hier is WAB 'n deelversameling van SxOxA. Swak magtigings in WAB is eksplisiet swak magtigings. Hier moet egter reëls vir implikasies gedefinieer word. Magtigingsbewerkings in hierdie

model sluit Toetsing, Toekenning en Wegneming van die reëls gedefinieer in die verskeie basisse in.

In die A dimensie of toegangstipe dimensie is 'n beperkte reeks noembare aksies wat uitgevoer kan word tussen aktiewe en passiewe entiteite, waarvan 'n paar reeds bestaan vanaf gewone databasisstelsels, soos bv. LEES, SKRYF, BYWERK, SKRAPPING EN SKEPPING. Hierdie aksies is ook moontlik tussen objekte, asook tussen die metodes wat deur die objek self uitgevoer word.

In 'n veilige stelsel, moet regte toegeken word aan die aktiewe entiteite om hierdie aksies uit te voer op die passiewe entiteite. Dit word gewoonlik gedoen deur 'n stelselsekerheidsbeampte, maar kan ook in die geval van diskresionêre sekerheidstelsels deur die eienaar van die entiteite toegeken word. Die regte word dan byvoorbeeld afgedwing deur die sekerheidstelsel alleen of die sekerheidstelsel in samewerking met 'n betroubare rekenbasis (BRB).

Larrondo-Petrie et al[Lar90] se beskermingsvereistes is weer gebaseer op die objekklas, en spesifiseer dat 'n gebruiker wat toegang besit tot 'n objekklas, dieselfde tipe toegang moet besit tot die ooreenstemmende subklasse se attribute wat geërf word vanaf die objekklas. Toegang word geweier tot die ander attribute in die subklas, indien anders gespesifiseer. Daar word egter ook gespesifiseer dat toegang na 'n volledige objekklas, toegang impliseer na die attribute van daardie objekklas sowel as na die attribute wat geërf was vanaf hierdie objekklas se ouerklasse. Indien daar meer as een voorouer is, is daar toegang tot die vereniging van die geërfde attribute.

Ons kan die volgende gevolgtrekking maak uit hierdie beginsel van Larrondo-Petrie et al.[Lar90] nl. dat toegang meer word met veralgemening(ouer klasse), terwyl toegang minder word met spesialisering(subklasse). D.w.s. toegang word meer waar minder inligting

beskikbaar is en minder waar meer inligting beskikbaar is. Hierdie beginsel kan gebruik word saam met diskresionêre of verpligte sekerheid, klassifikasievlakke kan bv. toegeken word aan gebruiker en dan mag laer klassifikasie slegs hoër objekvlakke(ouer objekklasse, dus veralgemenings) gebruik, terwyl gebruikers met hoër klassifikasievlakke laer objekvlakke(subklasse, of spesialiserings) gebruik.

In diskresionêre sekerheid kan die eienaar[Lin89] van 'n objekklas,objek, of objek subklas die reg tot hierdie beskermde entiteit verskaf. Verskillende vlakke van regte kan aan verskillende gebruikers toegeken word, volgens die diskresie van die eienaar.

'n Ander voorbeeld van hierdie tipe magtigingsmeganisme word gebruik in die modelle DAMOKLES en SODA, wat onderskeidelik diskresionêre en verpligte sekerheid implementeer. Die twee modelle word vervolgens bespreek.

5.4.4.2. DISKRESIONêRE SEKERHEIDSTOEGANGSBEHEER-MEGANISME.

In diskresionêre sekerheidstelsels besit die eienaar van die passiewe entiteite die reg om toegangsbeheer uit te oefen op sy entiteite, maar die stelselsekerheidsbeampte kan ook gebruik word in die afdwinging van hoër vlak sekerheid.

Die eienaar van die passiewe entiteite het die reg om die "reg om toegangsreg te beheer" uit te deel aan ander subjekte of gebruikers, maar terselfdertyd kan hy dit ook wegneem vanaf ander subjekte. Let op in DAMOKLES dat die subjek (Gebruiker,program) nie die eienaar van 'n objek is nie, maar die teendeel naamlik net die gebruiker of gebruikersgroep. Hierdie twee opsies veral, vereis 'n wye verskeidenheid sekerheid- of toegangsreëls om dit effektief en foutvry te hou. In DAMOKLES byvoorbeeld word dit vasgestel dat daar slegs een eienaar van

'n objek kan wees, d.w.s. in die geval van gebruikersgroepe word die groep administreerder die eienaar van die objek.

In DAMOKLES werk toegangsbeheer soos volg:

(a) Elke toegang moet altyd ten minste twee toegangsregte besit, een vir die gepaste databasis waartoe toegang benodig word en een vir die objekte waartoe toegang vereis word.

(b) As die bewerking wat uitgevoer word net die beskrywende gedeelte van die objek beïnvloed, is dit genoegsaam om die toepaslike reg te besit, maar as die bewerking ander gedeeltes van die objek soos die strukturele, weergawe of rol gedeelte beïnvloed moet daar meervoudige regte bestaan. Daar sal byvoorbeeld in sulke gevalle toegang tot die superobjek van hierdie objek benodig word.

(C) As voorbeeld van toegangsregte benodig gebruik ons kopiëring van objekte , hier is die volgende van toepassing : as 'n subjek 'n objek wil kopieer, het hy eerstens leesregte nodig vir die databasis waarin die objek is, asook vir die objek self, en daarna skryfregte vir die databasis waarheen die objek gekopieer gaan word.

Voordat daar nou in meer besonderhede na die toegangsbeheermeganisme gekyk word, moet daar eers gekyk word na die tipes bewerkings wat kan gebruik word tussen die passiewe en aktiewe lede van die sekerheidstelsel. DAMOKLES gebruik die volgende bewerkings :

(a)die BESTAAN-klas, wat die lees van databasissleutels van -objekte en verwantskappe omvat om te kyk of dit bestaan,

(b)die LEES-klas, wat die lees van databasisse, objekte (asook die attribute van die objekte) en verwantskappe omvat,

(c)die SKRYF-klas, wat die skryf van databasisse (invoeg, bywerk en skrap van objekte en verwantskappe), objekte (bywerk van attribute, invoeging van nuwe komponente en verwydering van komponente) en verwantskappe (bywerking van attribute en rolle), en laastens

(d)die SKRAP-klas, wat die skrapping van databasisse, objekte en verwantskappe omvat.

In DAMOKLES word hierdie bewerking klasse so opgedeel dat die regte tot een klas regte tot die volgende klas impliseer, soos volg :

BESTAAN < LEES < SKRYF < SKRAP.

Nadat die bewerkings wat moontlik is, vasgestel is, indien dit nodig was dat dit gespesifiseer moes word, moet daar nou toegangsreëls gedefinieer word wat toegang tot objekte sal beheer met inagneming van die bewerkings wat daarop van toepassing kan wees. Vir DAMOKLES is 'n toegangsreg die kombinasie van 'n bewerkingsklas en 'n beskermde objek, d.w.s. 'n toegangsreg R = (O,CO) as O die beskermde objek is, en CO die bewerkingsklas is. Indien daar egter nie 'n spesifieke lys van bewerkings bestaan waarom toegangregte gespesifiseer word nie, dan word toegangsregte met inagneming van ander tipes faktore gespesifiseer, soos byvoorbeeld in DISCO wat later bespreek word.

5.4.4.2. VERPLIGTE SEKERHEIDSTOEGANGSBEHEER-MEGANISME.

Multivlak of verpligte sekerheid beskerm passiewe entiteite wat geklassifiseer is op meer as een vlak, en laat deling tussen gebruikers of aktiewe entiteite met verskillende klaringsvlakke toe[Kee89]. Meestal word die passiewe entiteite gemerk met hul sensitiwiteitsvlakke en die aktiewe entiteite geklassifiseer met verskillende klaringsvlakke. 'n Multivlak of verpligte sekerheid stelsel arbitreer alle toegang van aktiewe entiteite na passiewe entiteite. Dié arbitrering word gewoonlik gedoen deur 'n verwysingsmonitor volgens 'n sekerheidsbeleid[Kee89], wat in die geval van SODA 'n betroubare rekenbasis(TCB) is.

Keefe et al[Kee89], noem verskeie probleme wat kan voorkom met multivlak sekerheidsdatabasisse, naamlik :

(A)Betroubaarheid : Die feit dat daar 'n groot aantal komplekse passiewe entiteite hanteer word met komplekse semantiese betekenis, plaas 'n groot las op die toegangsmonitor om te verseker dat alle toegange gemagtig is,

(B)die feit dat alle passiewe entiteite volledig en konsistent geklassifiseer moet word, en

(C)die voorstelling en manipulering van passiewe entiteite wat data bevat van meervoudige sensitiwiteitsvlakke, en laastens

(D)die interverwantskappe van die data en hul sematiek lei tot afleidingsprobleme. Afleiding kom voor as inligting wat vanaf die databasis herwin kan word, toelaat dat ander data afgelei kan word. Afleiding voorsien 'n vloei van data wat nie gearbitreer word deur die verwysingsmonitor nie.

In multivlak sekerheidsstelsels moet die *-eienskap uitgeoefen word, en spesifiseer dat passiewe entiteite met dieselfde of 'n hoër klassifikasievlakke boodskappe aan mekaar kan stuur, of van mekaar af kan ontvang, andersins

nie[Pfl89]. SODA[Kee89], byvoorbeeld, spesifiseer dat, omdat boodskappe aan klasse gestuur word en die voorkoms van daardie klasse (wat die metodes uitvoer) verskillende sensitiwiteitsvlakke kan besit, moet die klas objekte of betroubaar wees om hierdie multivlak objekte te hanteer, of daar moet 'n weergawe van die klas wees vir elke sensitiwiteitsvlak.

In SODA word toegang soos volg beheer :

Veronderstel dat 'n metode-aktivering uitgevoer word met 'n klaringsvlak van $L_{Sklaring}$ en 'n huidige klassifikasie van $L_{Shuidig}$ en toegang word gevra tot 'n gemerkte objek wat gestoor word in 'n gleuf met 'n sensitiwiteits beperkingsreeks van $[L_{onder}, L_{bo}]$. Die metode-aktivering word dan net toegelaat om

(a)die waarde van die gemerkte objek met sensitiwiteitsvlak L_o te lees as $L_o <= L_{Shuidig}$ is. 'n Onleesbare objek stuur 'n nil-waarde terug;

(b)'n gemerkte objek met 'n sensitiwiteitsvlak van $L_o = L_{Shuidig}$ te skep of te stoor in die beperkte gleuf as $L_{onder} <= L_{Shuidig}$ en $L_{Shuidig} <= L_{top}$; andersins word die bywerking geweier. Die klaringbeperking maak toelating dat die "SKRYF NA BO"-kondisie nie voorkom nie[Kee89], en verhoed so poliïnstansiëring as die klassifikasiereeks van die data 'degenerate' is, i.e. $L_{onder} = L_{top}$.

In SODA is die aktiewe entiteit, die subjek gesamentlik met die metode wat geaktiveer word. 'n Metode-aktivering word uitgevoer met 'n sekerheidsklassifikasievlak $L_{shuidig}$ bepaal deur twee faktore, die klaringsvlak $L_{sklaring}$ van die gebruiker en die tweede die huidige sekerheidsklassifikasievlak $L_{soorspronklik}$ van die metode-aktivering wat begin is deur 'n boodskap te stuur.

Reëls word gespesifiseer vir spesifieke gevalle wat mag voorkom, bv. in reël 1 is die aanteken reël, wat spesifiseer dat 'n metode begin met klassifikasievlak van $L_{shuidig}$ = Stelsel Laag(Laagste klassifikasievlak moontlik vir hierdie subjek, gespesifiseer in subjek se klassifisering). Reël 2 spesifiseer dat 'n metode-aktivering begin met 'n klassifikasievlak van $L_{shuidig} = L_{soorspronklik}$. Reël 3 spesifiseer dat as 'n gemerkte objek met 'n sensitiwiteitsvlak lo sodanig dat $L_{shuidig} <= L_o$ gelees of by 'n gepoliïnstansieerde versameling gevoeg word, is die klassifikasievlak van die metode die bogrens van $(L_{shuidig}, L_o)$. Hierdie is net 'n paar van die reëls as voorbeeld van hoe die reëls gespesifiseer word.

Die objek teruggevoer deur 'n metode-aktivering word gemerk met die metode-aktivering se klassifikasievlak $L_{shuidig}$.

Hierdie reëls verseker dat $L_{shuidig}$ altyd die vlak van die inligting beskikbaar tot die aktivering domineer. Die huidige klassifikasievlak begin by die laagste moontlike vlak om toe te laat dat die metode-aktivering die mees buigbare is. Die huidige klassifikasievlak bly behoue gedurende die metode-aktivering. As die metode terugkeer verdwyn die inligting geënkodeer in dié toestand van die aktivering. Die roeper gaan dan voort met sy oorspronklike huidige klassifikasievlak.

'n Boodskap word gestuur deur 'n metode-aktivering m_1 na 'n passiewe objek wat 'n ander metode-aktivering skep, m_2. M_1 voer uit met klaringsvlak $L_{sklaring}$ en 'n huidige klassifikasievlak van $L_{shuidig}$. Uit reël twee sê hulle metode-aktivering m_2 begin met dieselfde huidige klassifikasievlak en dieselfde klaringsvlak. Enige inligting wat oorgeplaas word na m_2 deur sy uitvoering te begin, is aanvaarbaar, want beide word met dieselfde klassifikasie uitgevoer.

In SEAVIEW en TRUDATA[And89] word sensitiwiteitsvlakke gevarieer om die sensitiwiteitsvlak van die data in 'n houer te beheer. TRUDATA gebruik dit om vatbaarheid te beperk, terwyl SEAVIEW dit gebruik om poli-instansiering te beperk, waar dit uitgebrei moet word met sekerheids beperkings vir veeltalmerking[And89].

D.w.s. in hierdie model was die vereistes eerstens dat multivlaksekerheid geïmplementeer moet word, waaruit gevolg het dat entiteite wat beskerming moet ontvang, asook entiteite wat die beskermde entiteite gebruik, klassifikasievlakke asook sensitiwiteitsvlakke moet ontvang. Dit kompliseer die werking van die model en moet in ag geneem word as die reëls vir die afdwinging van die sekerheid geskep word.

5.4.5. DIE VERMOË AS MAGTIGINGSMEGANISME

In DISCO weer is dit nie nodig dat spesifieke bewerkings gespesifiseer word nie, want daar word gebruik gemaak van 'n VERMOë(onvervalsbare teken wat die bewerker in staat stel om toegang tot 'n verwante passiewe entiteit te verkry). Hierdie vermoë word gekoppel aan 'n boodskap en dien as sleutel tot die passiewe entiteit wat gebruik word, hetsy 'n objek, objekklas of voorkomsveranderlike. Indien die aktiewe entiteit wat die objek, objekklas of voorkomsveranderlike wil gebruik, nie die toepaslike vermoë vir daardie spesifieke passiewe entiteit het nie, beteken dit geen toegang tot die passiewe entiteit nie. Eienaarskap word nie net toegeken op die objekvlak nie, maar ook in fynere vlakke soos metodes en voorkomsveranderlikes.

In die meeste diskresionêre sekerheidstelsels word diskresionêre sekerheid gekombineer met 'n hoeveelheid verpligte sekerheidstoegangsreëls, omdat dit die veiligheid van die sekerheidstelsel verfyn. DISCO is 'n voorbeeld hiervan, want daar word sensitiwiteitsvlakke aan die entiteite toegeken, en byvoorbeeld gesê dat as een entiteit (e_1) 'n sensitiwiteitsvlak het wat hoër is as 'n ander s'n

(e_2) moet die subjekte wat toegang het tot e_1 ook toegang besit tot e_2. 'n Voorbeeld hiervan wat in die meeste modelle voorkom, is dat 'n aktiewe entiteit slegs toegang verkry tot 'n objek of subklas hierdie aktiewe entiteit reeds toegang besit tot die superklas van hierdie objek. D.w.s. in DISCO word toegangsreëls gespesifiseer met inagneming van die sensitiwiteitsvlak van die passiewe entiteit en die vermoë wat besit moet word vir hierdie passiewe entiteit.

GEVOLGTREKKING

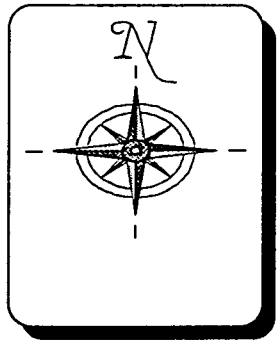
In hierdie hoofstuk het ons nou die stappe saamgevat wat u kan gebruik in die bou van 'n sekerheidsmodel, en ons het ook riglyne gegee wat u kan gebruik in die keuses wat gemaak kan word in die verskeie stappe. Daar is veral gekyk na die uiteensetting van 'n model, die beleidsrigtings wat ingebou kan word in die model en die elemente van 'n tipiese model. Die meeste van die keuses is ook beïnvloed deur die mate van sekerheid wat benodig word in die sekerheidsmodel. Diskresionêre en verpligte sekerheid speel veral 'n rol in die mate van sekerheid van 'n stelsel en daarom is daar deurentyd daarna verwys.

Die volgende hoofstuk bied nou 'n nuwe sekerheidsmodel wat poog om sekere van die tekortkominge in vorige generasie databasissekerheidsmodelle te oorbrug.

HOOFSTUK 6 - DIE DISKRESIONêRE SEKERHEIDSMODEL

(DISMOD)

Daar is nou voldoende agtergrondinligting gegee dat die nuwe model voorgestel kan word. DISMOD is 'n diskresionêre sekerheidstelsel vir alle objekgeoriënteerde omgewings. DISMOD se doel is om 'n diskresionêre sekerheidsmodel daar te stel wat 'n buigbare sekerheidsoplossing bied vir alle objekgeoriënteerde databasisomgewings.



Die volgende drie hoofstukke bied die ontwikkeling van DISMOD soos volg aan: hoofstuk ses sal aan u die elementêre beginsels of buitelyne van DISMOD voorstel, waarna hoofstuk sewe die werkinge van die model in breë trekke aan u sal uitlê en laastens sal hoofstuk agt vir u die verdere implikasies van hierdie model na vore bring.

6.1. DOELWIT VAN DISMOD

DISMOD is 'n diskresionêre sekerheidstelsel. Die doelwitte van die sekerheidstelsel kan soos volg saamgevat word:

- Die sekerheidstelsel moet alle volgende generasie databasisstelsels kan ondersteun, maar in besonder die objekgeoriënteerde databasisomgewing.
- Die stelsel maak grootliks gebruik van diskresionêre sekerheid, maar kan ook gebruik maak van verpligte sekerheidsmeganismes in kombinasie met diskresionêre sekerheid om die mees doeltreffende sekerheidsmeganisme daar te kan stel.

- Die sekerheidstelsel poog om 'n baie fyner grein van beheer te kan uitoefen, deurdat sekerheid toepasbaar is tot in die fynste besonderhede van alle elemente in die stelsel.
- Die model moet buigbaar, aanpasbaar en hoogs betroubaar wees.
- Konfidensialiteit, integriteit en geheimhouding van die stelsel elemente moet te alle tye as baie belangrik beskou word.
- Die sekerheidsmodel self maak gebruik van objekgeoriënteerde konsepte in die uitoefening van die bogenoemde vereistes.

Die doel van DISMOD is dus baie duidelik, naamlik dat dié sekerheidsmodel 'n fyner grein van sekerheid wil implementeer in objekgeoriënteerde omgewings. Die ideale basis vir die sekerheidsmodel is dan ook 'n objekgeoriënteerde databasis, oftewel die gebruik van die objekgeoriënteerde programmering.

Die kenmerke of ***karaktertrekke*** van die sekerheidsmodel kan soos volg saamgevat word:

- Beskerming word verskaf vir entiteite in 'n omgewing op al die vlakke van die entiteit. Veronderstel bv. dat die entiteit 'n objek is. In hierdie geval word beskerming verskaf vir die metodes of die gedrag van die objek, die eienskappe of data van die objek en ook vir die objek in geheel. Indien die entiteit 'n klas is, word beskerming gebied vir die klas self en alle subklasse en objekte in die klas.
- Beskerming word verskaf volgens die diskresie van die **eienaar** van 'n objek, wat 'n eienskap van 'n ware diskresionêre sekerheidsmodel is.
- Daar word gebruik gemaak van 'n stelselsekerheidsbeampte om die oorhoofse stelselveiligheid te waarborg. Dié funksie kan ook deur 'n databasisadministrateur uitgeoefen word.
- Rol-gebaseerde sekerheid kan gebruik word indien regte uitgedeel word. 'n Voorbeeld hiervan is wanneer regte om sekere entiteite te gebruik, uitgedeel word volgens die rol van die betrokke subjekte wat die regte ontvang.

- Entiteite word só beskerm dat dit nie moontlik is vir enigeen om die entiteite te gebruik sonder die besit van die toepaslike sleutel van die entiteite nie.
- 'n Entiteit kan dus slegs gebruik word indien die een wat dit wil gebruik in besit is van 'n vermoë (nie-vervalsbare teken of sleutel) vir die entiteit wat hy wil gebruik.

Noudat u weet wat die doel en kenmerke van die model is, kan ons begin kyk na die komponente van die model. Die volgende afdeling sal hierdie komponente in meer besonderhede beskou.

6.2. KOMPONENTE VAN DISMOD

Die komponente van DISMOD is die volgende:

A Die ***entiteit*** wat beskerm word,

B Die ***subjekte*** waarteen of waarvoor ons die entiteite beskerm

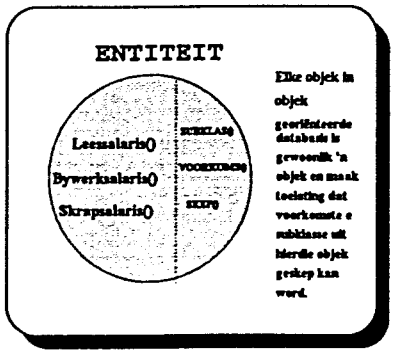
C Die ***eienaar*** wat sy entiteite beskerm,

D Die ***vermoë*** as sleutel tot beskermde entiteite

E Die ***stelselsekerheidsbeampte*** as oorhoofse sekerheidsbevestiger en ook publieke domeinskepper.

F Die **vermoëlyste, en gesentraliseerde vermoëlys.**

Elk van die bogenoemde komponente word vervolgens bespreek..



6.2.1. DIE ENTITEIT

'n Entiteit in DISMOD is enige element van die objekgeoriënteerde databasis. Veronderstel bv. die maatskappy het 'n personeelafdeling. Die afdeling sal tipies 'n personeeldatabasis besit waarin alle besonderhede van hul

werknemers gestoor word. Die personeeldatabasis besit dan ook gewoonlik 'n salaris, pensioen en naam-en-adres komponent.

Veronderstel die objekgeoriënteerde databasis lyk nou soos volg:

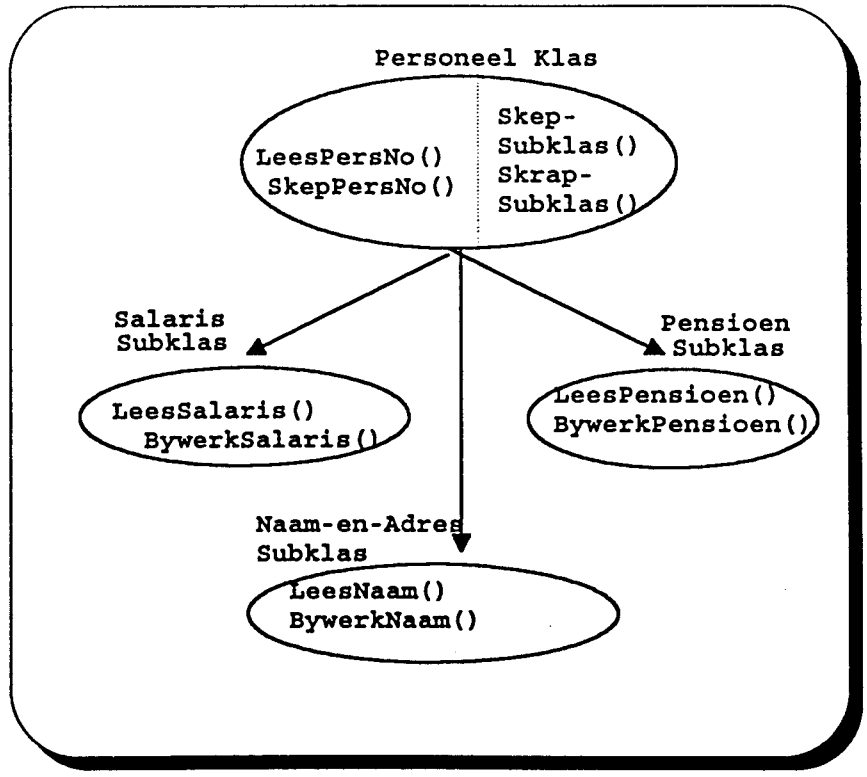


Fig 6.1. Voorbeeld Databasis wat beskerm word.

In die databasis in Figuur 6.1 is elk van die entiteite wat u kan sien, die Personeel Klas, en die Salaris-, Pensioen- en Naam-en-Adres-subklasse objekte. Objekte besit beide gedrag en data (eienskappe). Die objek sal in geheel beskerm word, maar ook die kleiner vlakke, sy gedragspatrone of metodes en sy eienskappe sal byvoorbeeld ook beskerming geniet.

Dit is dus nou duidelik wat ons gaan beskerm. Laat ons dan nou voortgaan en aan u voorlê wat die subjekte of gebruikers in DISMOD is.

6.2.2. SUBJEKTE

Subjekte is alle gebruikers of gebruikersgroepe wat die entiteite in die stelsel gebruik deur aan die entiteite boodskappe te stuur. Die stuur van boodskappe

inisieer die aksies of gedragspatrone van dié betrokke entiteit waaraan die boodskap gestuur is. Die objek reageer op sy beurt met 'n antwoord of die uitvoering van 'n aksie.

'n Subjek kan ook 'n objek wees wat 'n boodskap aan 'n ander objek stuur. Die geroepte objek sal weer eens 'n antwoord terugstuur of 'n aksie uitvoer, afhangende van die versoek.

In 'n diskresionêre sekerheidstelsel is daar unieke subjekte wat die reg besit om beheer oor spesifieke entiteite uit te oefen. Die subjekte staan bekend as ***eienaars*** en moet tipies die volgende eienskappe besit:

A Die eienaar van 'n entiteit is die subjek wat die entiteit geskep het

B Indien 'n entiteit geskep word, verkry die subjek **eienaarskap** oor die entiteit wat beteken dat die subjek die reg het om hierdie entiteit te gebruik. Dit beteken ook dat die eienaar die **REG** om die entiteit te kan gebruik, kan uitdeel aan ander subjekte.

C Die reg wat die eienaar aan ander subjekte gee om die entiteit te kan gebruik, is gewoonlik in die vorm van 'n ***sleutel*** tot die entiteit. Dié sleutel kan gevorm word om te pas by die subjek wat die sleutel ontvang. Die personeelklerk kan byvoorbeeld 'n sleutel tot die salarissubklas verkry maar hierdie sleutel gaan hom net toelaat om salarisse te lees en nie by te werk nie. Sleutels van hierdie vorm, staan deurgaans bekend as vermoëns

D Die eienaar van 'n entiteit besit ook die **reg** om enige sleutel wat hy aan ander subjekte uitgedeel het weer terug te neem.

E Dit is belangrik om daarop te let dat die veiligheid van hierdie sekerheidsmodel rus op die skouers van die ***eienaars*** van entiteite om die entiteite wat aan hul behoort volgens hul diskresie te bestuur.

Die subjek, asook die eienaar, is nou bekend gestel. Daar is gebruik gemaak van die term sleutel tot 'n entiteit. Die sleutel tot 'n entiteit

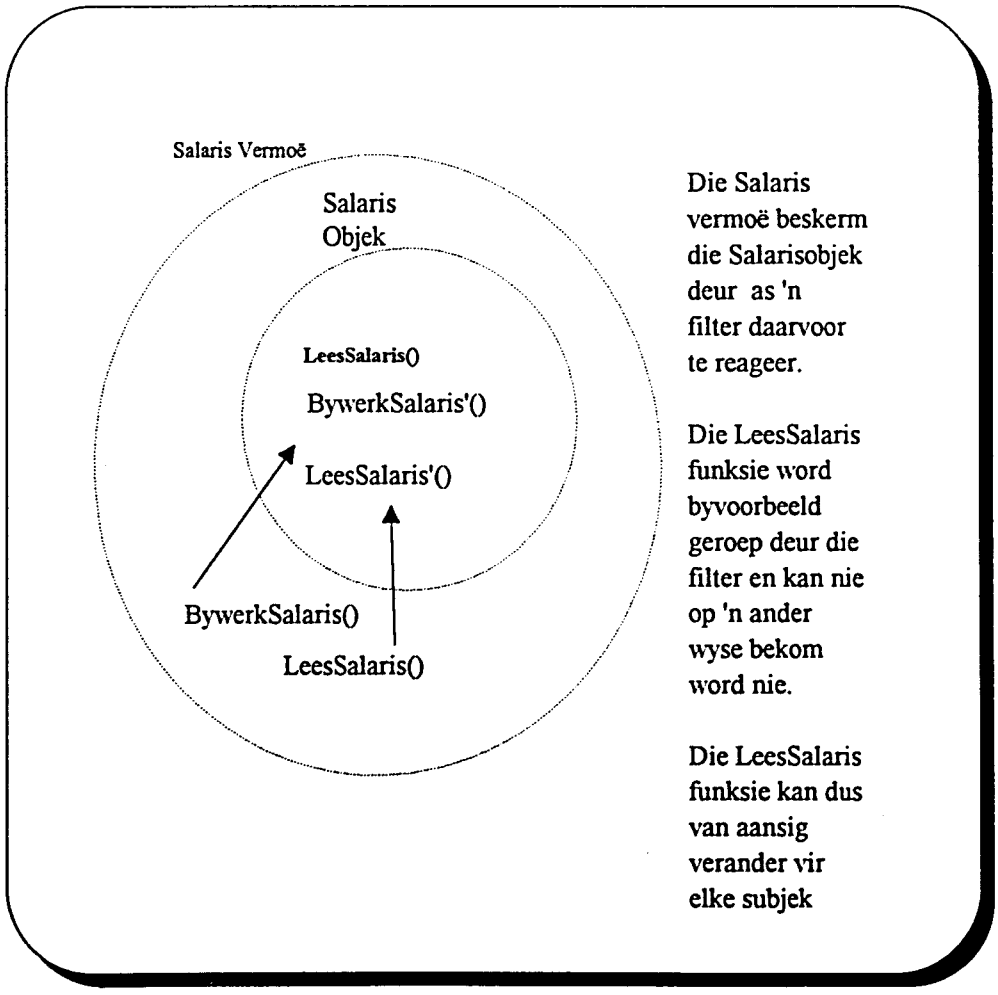


Fig 6.2. Die Vermoë as sleutel

word ook 'n ***vermoë*** genoem en is 'n nie-vervalsbare identifiseerder wat gebruik word om 'n entiteit te identifiseer in tradisionele terme. Let veral op figuur 6.2. om die gebruik van die vermoë as filter te verstaan. In die loop van hierdie model sal die vermoë as 'n ***nie-vervalsbare identitfiseerder*** gebruik word wat die entiteit sal oopsluit op die voorgeskrewe wyse.

6.2.3. DIE VERMOë

'n Vermoë is 'n objek met karakter en gedrag. Dit dien as 'n filter tot 'n entiteit wat 'n nie-vervalsbare identifiseerder is vir die entiteit. Die vermoë lyk soos volg :

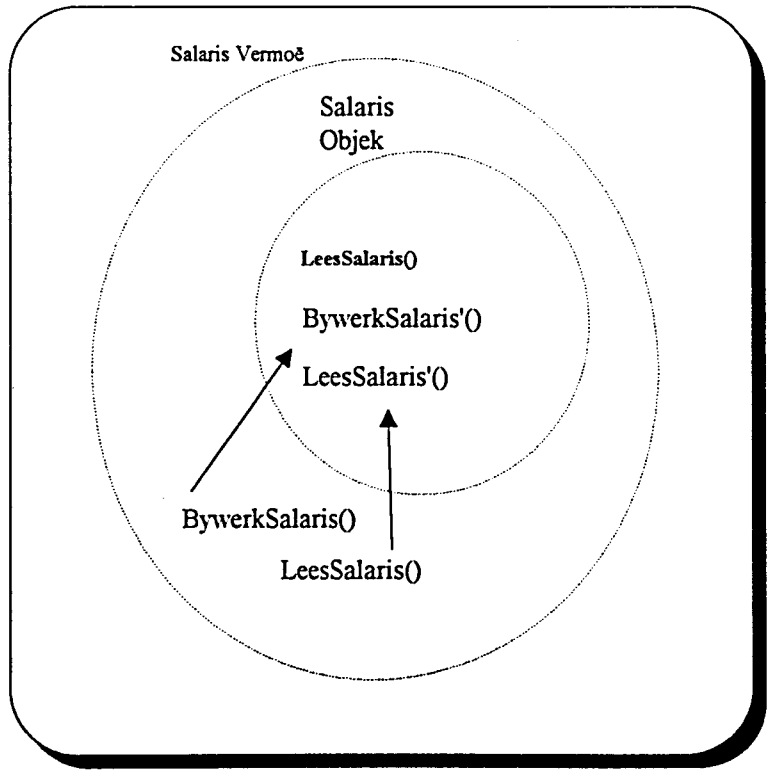


Fig 6.3 Die Vermoë as Sleutel.

Let veral op die werking van die vermoë soos geïllustreer en verduidelik in figuur 6.2.

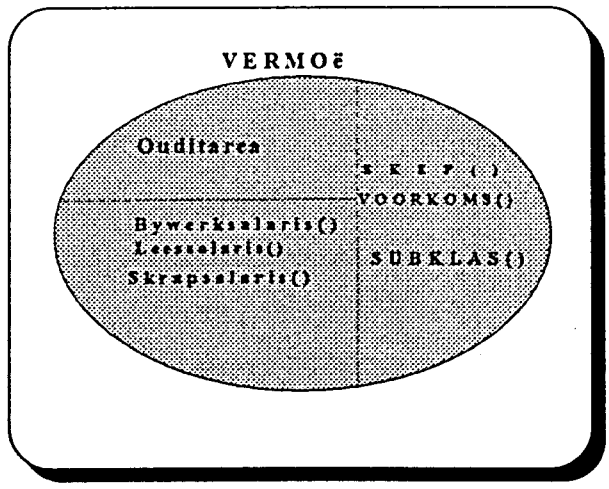
Soos u kan sien, bied die LeesSalaris() en BywerkSalaris() funksies in die Salarisvermoëobjek die sleutel tot die Salarisobjek. Dié funksies in die salaris vermoë bevat nie die instruksies om die taak uit te voer nie, maar bevat eerder die identifiseerder of unieke sleutels tot die funksie in die salarisobjek om daardie spesifieke funksie te roep om homself uit te voer.

'n Vermoë kan net bekom word indien 'n objek geskep word en word dan in die skepproses die besitting van die eienaar van daardie objek. 'n Objek kan egter nie geskep word nie, tensy die subjek wat die entiteit wil skep, 'n Skepvermoë besit.

Vermoëns kan ook hierargieë vorm. Die eienaar van 'n entiteit sal saam met 'n vermoë as sleutel tot sy objek, 'n vermoëklas ontvang, wat hy kan gebruik om ander sleutels of vermoëns te ontwikkel as sleutels tot sy entiteit.

Dié vermoëklasse word uitgebrei deur verskillende subklasse en voorkomste te vorm wat verskillende tipes toegange tot die entiteit definieer. Voorkomste van die vermoëklasse kan uitgedeel word aan ander subjekte om as sleutels tot die eienaar se entiteit te dien.

Die feit dat die voorkomste en vermoësubklasse uitgedeel kan word aan ander subjekte, maak dit noodsaaklik om 'n ouditarea (sien fig 6.4) te hê waar rekord gehou word van die subjekte aan wie hierdie vermoëns uitgedeel is.



Die vermoë lyk soos volg:

Fig 6.4 Die Vermoë met Ouditarea

Die Ouditarea van 'n vermoë is 'n bergingsarea waarin die name of identifiseerder van die subjekte - waaraan regte of vermoëns tot die eienaar se entiteit toegeken is - gestoor word. Die ouditarea word net bygewerk op die vlak waar die vermoëns uitgedeel word. Beskou die volgende voorbeeld.

Veronderstel die personeelbestuurder is die eienaar van die personeeldatabasis. As hy 'n vermoë tot die personeeldatabasis toeken of uitdeel aan sy hoofklerk, word daar in die vermoëklas 'n inskrywing gemaak dat die hoofklerk 'n vermoë besit tot die personeeldatabasis en ook watter vermoë hy besit. Veronderstel verder dat hy 'n uitdeelreg aan die hoofklerk gee, d.w.s.

'n reg dat die hoofklerk ook vermoëns kan uitdeel tot hierdie personeeldatabasis, dan sal daar in die vermoësubklas wat die hoofklerk besit, rekord gehou word van die subjekte aan wie die hoofklerk vermoëns uitdeel.

Die feit dat die vermoësubklas van die hoofklerk deel is van die vermoëklas van die personeel bestuurder, stel die personeel bestuurder in staat om te sien aan wie sy hoofklerk vermoëns uitgedeel het tot sy entiteit.

Die ouditarea van 'n vermoëklas of -subklas speel dus 'n belangrike rol in die veiligheid of konfidensialiteit en integriteit van elke entiteit.

Vermoëns en vermoëklasse word gesamentlik gehou in 'n beskermde sekerheidstelsel, wat gehou word in 'n beskermde area in die geheue. Dit is dus nie moontlik om hierdie stelsel maklik te kan verander nie. Subjekte is in der waarheid nie in besit van die werklike vermoë as objek of vermoëklas as objekklas nie, maar slegs 'n identifiseerder van só 'n vermoë of vermoëklas. Dié identifiseerders word gehou in 'n **vermoëlys** van die subjek. Die vermoëlys van die subjek kan ook net wysers bevat na 'n **gesentraliseerde vermoëlys** waarin die identifiseerders van die betrokke vermoëns of vermoëklasse gehou word. Vermoëlyste en die gesentraliseerde vermoëlys word ook in 'n beskermde area van die geheue gehou.

'n Geheelbeeld van die rekenaaromgewing om meer duidelikheid te gee aan die bogenoemde beskrywing kan in fig 6.5. gesien word . Die sekerheidstelsel word beskerm in die geheue en maak gebruik van die gesentraliseerde vermoëlys, wat ook beskermd is. Die gesentraliseerde vermoëlys word gebruik met elke versoek wat gerig word aan die rekenaarstelsel om die korrekte metodes te vind om uit te voer. Elke versoek word getoets om te sien of die sender van die versoek die betrokke vermoë besit in die subjek se vermoëlys voordat die versoek uitgevoer word. Die versoek word uitgevoer deur die betrokke vermoë as filter te gebruik vir die entiteit waarnatoe die versoek gestuur word. Die vermoë (filter) verbind die

versoek met die korrekte metode vir die versoek en inisieer dan die uitvoering van die metode.

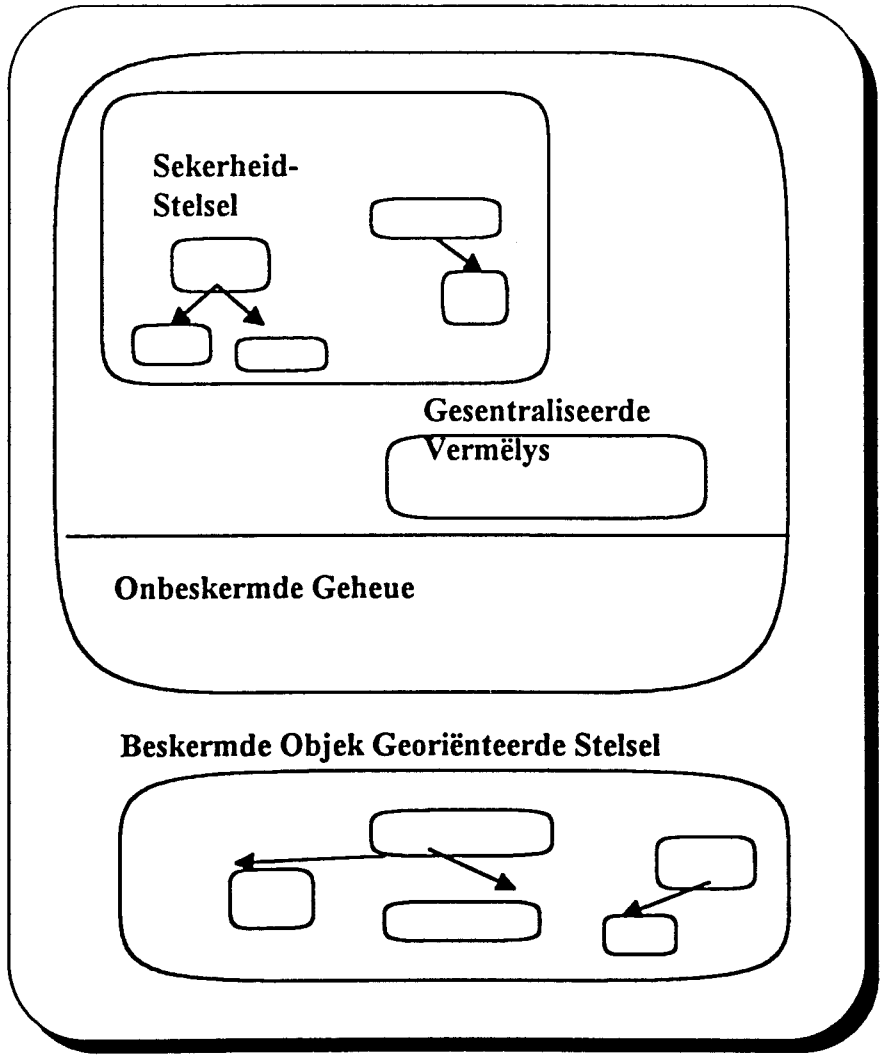


Fig 6.5. Die Rekenaar Omgewing

Vroeër is gemeld dat 'n subjek nie 'n entiteit sonder 'n skep vermoë kan skep . Die skepvermoë word aan 'n subjek toegeken deur die stelselsekerheidsbeampte. Die stelselsekerheidsbeampte moet egter nou eers voorgestel word voordat verder gegaan kan word.

6.2.4. DIE STELSELSEKERHEIDSBEAMPTE(SSB)

Die stelselsekerheidsbeampte is 'n subjek met die hoogste regte in die stelsel. Dié subjek is verantwoordelik vir die veiligheid van die stelsel, d.w.s.

vir die integriteit van eienaars. Let wel dat hierdie pos ook gevul kan word deur 'n databasisadministrateur.

Die Stelselsekerheidsbeampte het mag oor alle eienaars, d.w.s. hy besit die reg om enige vermoëns of sleutels wat in die besit van 'n eienaar is, weg te neem. Die doel van die SSB is om in uitsonderlike gevalle van oortreding te kan intree en daarom moet hy die mag of reg besit om enige vermoë van 'n subjek of eienaar te kan wegneem indien daar nie deur die eienaar korrek opgetree word nie.

Die stelselsekerheidsbeampte se funksies sal later in meer besonderhede bespreek word.

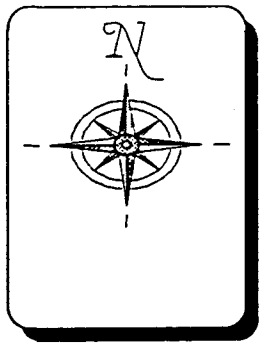
AFSLUITING

Die Model is nou in breë trekke verduidelik en dit behoort nou duidelik te wees dat die model 'n fyner grein van beheer bied. Die volgende hoofstuk sal die werking in meer besonderhede verduidelik, en die laaste hoofstuk sal die implikasies en komplikasies van die model duidelik uiteensit en oplossings daarvoor aanbied.

---oOo---

HOOFSTUK 7 -DIE WERKING VAN DISMOD

In hoofstuk ses is die doel en komponente van DISMOD voorgestel, maar die werking van DISMOD het nog heelwat bekendstelling nodig. Die werking van DISMOD sal in die volgende kategorieë verdeel en bespreek word, :



A Kontrole oor die skepping van entiteite, asook die werking van die skepping van entiteite

B Kontrole oor die uitdeling van vermoëns

C Kontrole oor die wegneem van vermoëns

D Die gebruik van entiteite sonder vermoëns (Is dit moontlik?)

E Beheermoontlikhede van die eienaar van 'n entiteit.

Die werking van die DISMOD sal met behulp van die personeelobjek-databasis (in die vorige hoofstuk) verder bespreek word.

7.2. BEHEER OOR DIE SKEPPING VAN ENTITEITE.

Entiteite is die elemente van die objekgeoriënteerde databasis wat beskerming nodig het. Dit is egter noodsaaklik dat die versameling entiteite wat beskerm word nie vervalste of onnodige entiteite insluit nie.

Die vraag wat seker op hierdie stadium na vore kom is die volgende:

"Is dit moontlik om entiteite te skep, sonder om die REG te besit om dié entiteite te skep, d.w.s. is dit moontlik dat enigeen enige entiteit kan skep?"

Die antwoord op hierdie vraag is nee want 'n subjek mag slegs 'n entiteit skep indien hy in besit is van 'n spesiale **skepvermoë** wat aan hom gegee is deur die stelselsekerheidsbeampte of die databasisadministrateur. Die Skepvermoë word **nét** uitgedeel deur die stelselsekerheidsbeampte of databasisadministrateur in die geval van 'n objekklas, oftewel die hoogste vorm van die betrokke entiteit. Dié vermoë besit 'n skepmetode wat die subjek in staat sal stel om spesifiek die entiteite te skep waarvoor hy aansoek gedoen het.

Met die skep van 'n vermoë sal die subjek verskeie beskermingselemente en onderskeidende eienskappe ontvang, waaronder die volgende :

A **Eienaarskap** van die entiteit wat aangedui word in die vermoëklas wat die subjek ontvang, asook deur die invoeging van die identiteit van die vermoëklasidentifiseerder en vermoëidentifiseerder in die vermoëlys van die subjek.

B 'n **Vermoë** wat toegang aan die subjek, as *eienaar*, bied tot die geskepte entiteit,

C 'n **Vermoëklas** wat gebruik kan word vir die definiëring van ander vermoëns en vermoësubklasse om spesifieke aansigte vir ander subjekte tot dié entiteit te verskaf.

Die feit dat die Stelselsekerheidsbeampte of databasisadministrateur 'n vermoë aan die subjek verskaf het om 'n entiteit te kan skep, maak hom die ***supereienaar*** van die entiteit, maar hierdie feit sal die eienaar van die entiteit nie weet nie. Dit stel egter die stelselsekerheidsbeampte of databasisadministrateur in só 'n posisie dat hy die vermoë van die eienaar tot die geskepte entiteit kan wegneem indien dit misbruik word of indien daar omstandigdhede was waarin die eienaar nie meer verder beheer kon uitoefen oor die betrokke entiteit nie..

Die gebruik van hierdie reg van die stelselsekerheidsbeampte om vermoëns weg te neem, sal egter net in uitsonderlike gevalle gebruik word. 'n

Voorbeeld van 'n situasie waar dit gebruik sal word, is wanneer 'n eienaar van 'n entiteit skielik sterf en slegs hy die reg gehad het om sy entiteit te beheer. In hierdie gevalle is dit noodsaaklik om die eienaarskap weg te neem en 'n nuwe eienaar aan te stel of alle skakels of toegangsregte van ander subjekte af weg te neem. Afhangende van die situasie kan die stelselsekerheidsbeampte besluit wat om met die eienaarskap van daardie spesifieke entiteit te maak.

Met die skep van entiteite sal subjekte skepmetodes as deel van die vermoë van die entiteit ontvang, indien dit moontlik is dat daardie entiteit uit onderafdelings kan bestaan, of indien dit moontlik is dat die entiteit opgebou kan word uit 'n hiërargie van ander entiteite.

'n Voorbeeld van die skep van 'n entiteit is as volg:

A. Die Personeelbestuurder doen aansoek by óf die stelselsekerheidsbeampte (SSB) óf die databasisadministrateur om 'n personeelobjekdatabasis te skep.
B. Die stelselsekerheidsbeampte(SSB) ondersoek dan die regte van die personeelbestuurder en kom dan (gestel) tot die bevinding dat hy wel die reg het om hierdie tipe databasis te skep.
C. Die SSB sal nou 'n vermoëvoorkoms uit die sekerheidstelsel vermoëklas skep met 'n spesifieke skepmetode wat die personeelbestuurder(subjek) in staat sal stel om die personeelobjekdatabasis te skep.
D. Die SSB gee nou hierdie vermoë aan die personeelbestuurder (subjek).
E. Die Personeelbestuurder gebruik dan die vermoë wat aan hom gegee is om die boodskap 'SKEP(Personeelobjekdatabasis) ' aan die beskermde stelsel te stuur. Dan word 'n Personeelobjekdatabasis geskep, asook die vermoë wat toegang bied tot hierdie Personeelobjekdatabasis en 'n vermoëklas vir die definiëring van die

aansigte van toegangsregte. Dié vermoë en vermoëklas word dan aan die personeelbestuurder gestuur.

Die Personeelbestuurder se vermoëklas besit nou 'n skepmetode wat hom sal toelaat om subklasse en voorkomste in sy personeel objek databasis te skep, omdat die entiteit wat geskep was, 'n KLAS was. Dit wil sê dat die vermoëklasse wat geskep word vir eienaars, só ontwikkel word dat dit die skep funksies wat moontlik is op die beskermde entiteit, insluit in die vermoëklas van die eienaar.

Die personeelbestuurder kan dus nou 'n pensioen, 'n salaris en 'n naam-en-adres-subklas in die personeelobjekdatabasis te skep. Daar sal in elk van hierdie gevalle vir elk van hierdie subklasse wat geskep word ook vermoëns en vermoëklasse geskep word en die eienaar sal geïdentifiseer word met die nodige identifiseerders in die vermoëlys van die eienaar. Die vermoëklasse en vermoëns wat geskep word, is alles deel van die vermoëklas van die ouerobjek, in hierdie geval die personeeldatabasis. Dit is in der waarheid subklasse van die vermoëklas. Die feit dat metodes en data geërf word vanaf ouerklasse, maak hierdie proses noodsaaklik. Die proses sal nou met 'n voorbeeld geïllustreer word.

Veronderstel eerstens dat die personeel objekklas, die volgende metodes bevat :

(a) Lees_Personeel_Besonderhede()
(Die funksie verskaf byvoorbeeld net die personeelnommers wat in die databasis bestaan.)
(b) Vind_Nuwe_Personeel_nommer()
(Die funksie verskaf byvoorbeeld 'n algoritme waarvolgens nuwe personeelnommers toegeken word.)

Die vermoë wat toegang tot die vermoë objekklas toelaat sal soos in figuur 7.1. gesien, vertoon.

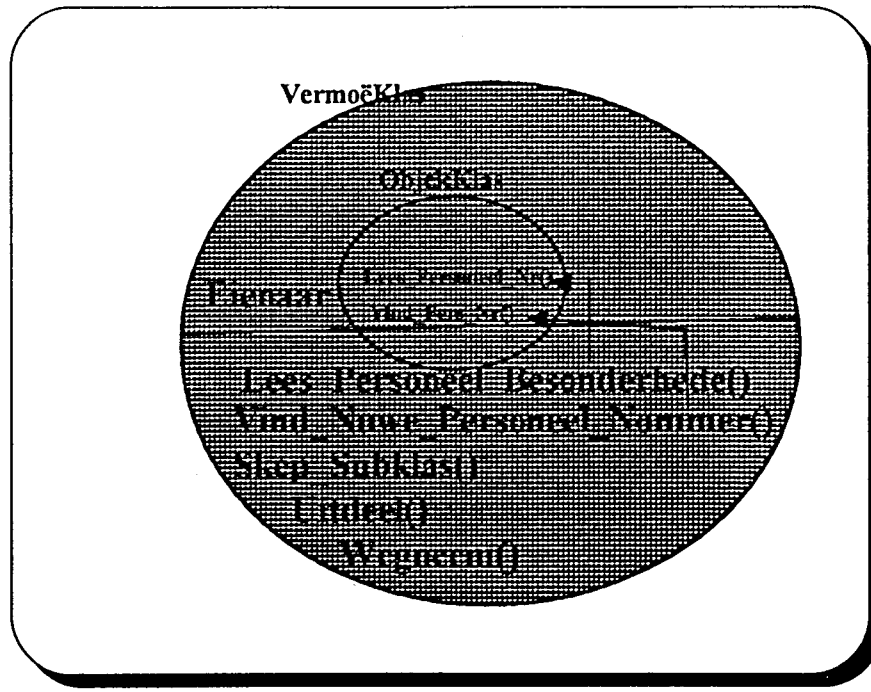


Fig 7.1. Voorbeeld van 'n VermoëKlas as filter

Die eerste punt waarop gelet kan word in fig 7.1. is dat die eienaar van die entiteit gemerk word in die vermoëklas. Die eienaar identiteit in die vermoëklas word geënkripteer en kan slegs verander word deur die skrapping van hierdie vermoëklas en die skepping van 'n nuwe vermoëklas.

'n Tweede punt waarop gelet kan word is dat boodskappe wat gestuur word deur subjekte, gebruik maak van die metode naam in die vermoë en nie die metode naam in die objekklas self nie, want dit is nie bekend aan die subjek nie.

'n Derde punt waarop gelet moet word, is dat hierdie figuur die vermoëklas as filter uitbeeld, en daarom is uitdeel, wegneem en skep funksies ingesluit in dié filter. Indien die vermoë wat geskep was gebruik was as filter het slegs die metodes van die entiteit daarin verskyn.

Die Skep_Subklas() funksie in die vermoëklas kan nou deur die subjek gebruik word om enige onderdele of subklasse of voorkomste in die objekklas (PersoneelKlas) te skep.

Veronderstel die eienaar van die personeel_Klas gebruik nou die skep_Subklas() funksie en stuur 'n boodskap skep_subklas(Salaris_subklas) aan die stelsel.

Die sekerheidstelsel sal die volgende aksies inisieer :

A 'n Subklas - Salaris - sal geskep word uit die personeel klas. Dit beteken eerstens dat alle metodes en data geërf word vanaf die personeelklas.

B Die subjek wat die boodskap gestuur het, wat slegs die eienaar van die ouer klas kon wees, ontvang die reg om die metodes en data van die nuwe subklas te definieer.

C Indien die subjek die definiëring voltooi het, skep die sekerheidstelsel 'n vermoësubklas met die identifiseerders van die metodes en data van die salaris subklas uit die vermoëklas van die ouerklas. Dit wil sê, alle metodes van die personeel-ouer-klas word geërf in die subklas. 'n Voorkoms of vermoë word ook van hierdie subklas geskep waarmee net die subklas se metodes en data bereikbaar is, d.w.s. daar is geen ekstra funksies soos uitdeel() funksies, ens. in bevat nie.

D Indien die eienaar 'n subklas wil skep waarin die ouer klas se funksies of metodes nie gebruik mag word nie, moet daar 'n nuwe subklas geskep word wat hierdie metodes as nul-funksies herdefinieer.

Die bogenoemde aksies se resultaat kan soos volg voorgestel word :

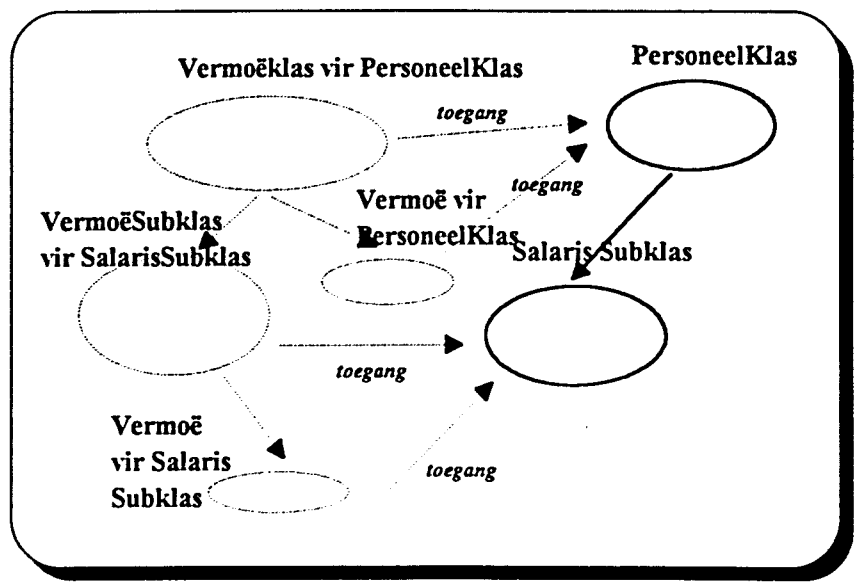


Fig 7.2. Die resultaat van die skep van 'n subklas.

Indien die Personeelbestuurder nie die skepping van die subklasse wou hanteer nie maar eerder hierdie funksies vir sy hoofpersoneelklerk wou gee om te doen, kon hy 'n vermoë aan die hoofklerk uitgedeel het wat die reg aan hom sou gee om hierdie entiteite te skep. Die feit dat die vermoë wat aan die hoofklerk uitgedeel word 'n deel is van die vermoëklas van die eienaar, maak dit moontlik dat alles wat nou geskep word, nog steeds onder beheer is van die werklike eienaar. Daar sal verder in die verloop van die hoofstuk na só 'n verhouding verwys word as 'n **EIENAAR-SUBEIENAAR** verhouding.

Die Hoofklerk kan nou voortgaan om met die vermoë wat hy ontvang het 'n pensioen- en naam-en-adres-subklas te skep. Die feit dat die hoofklerk, as subjek die entiteite geskep, het maak van hom die sub-eienaar van die entiteite en hy sal ook die vermoëns en vermoëklasse van die entiteite ontvang maar hierdie vermoëns en vermoëklasse sal subklasse van die eienaar se vermoëklas wees. Die hoofklerk sal dus die sub-eienaar van die eienaar wees.

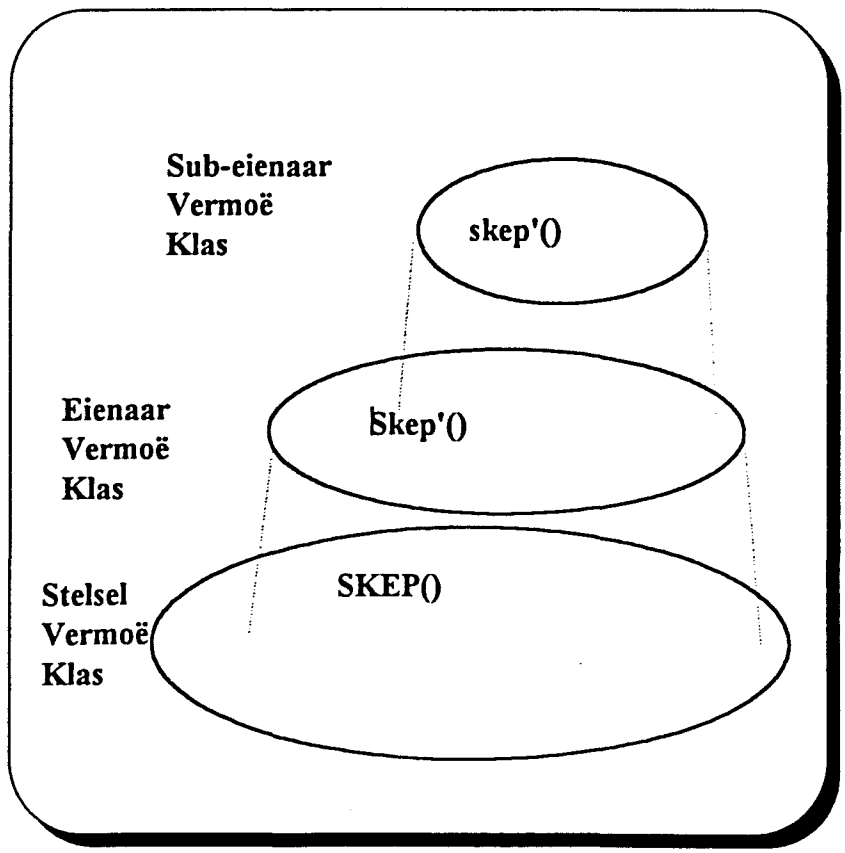


fig 7.3. Die hierargie van die "Skep"-vermoë

Figuure 7.3. illustreer hoe elke skepvermoë die subjek wat bo hom is die reg gee tot die beheer van sy entiteite.

Die feit dat die eienaar van 'n entiteit se naam in die vermoëklas of -subklas geënkripteer word, beteken dat slegs daardie subjekte die genoemde objekte mag verander. 'n Reël wat belangrik en voortaan sal geld, is die volgende :

Die eienaar van 'n ouerklas of vermoëklas mag te eniger tyd die vermoësubklasse en voorkomste van daardie betrokke ouerklas of superklas verander en besit ter alle tye toegang tot die subklasse.

Dit beteken dat die sekerheidstelsel slegs die eienaar van 'n objek, objekklas of objeksubklas sal toelaat om te verander daaraan, tensy die subjek wat dit wil doen die eienaar van die ouerklas of superklas van die betrokke objek is.

Uit die bogenoemde blyk dit dus duidelik dat die eienaar van 'n entiteit ten volle in beheer van sy entiteit is, alhoewel hy regte binne die konteks van sy entiteit kon uitdeel aan ander subjekte. Figuur 7.4. toon aan dat die ouditarea van die ouerklas ook te alle tye bygewerk word in die skepproses bygewerk word.

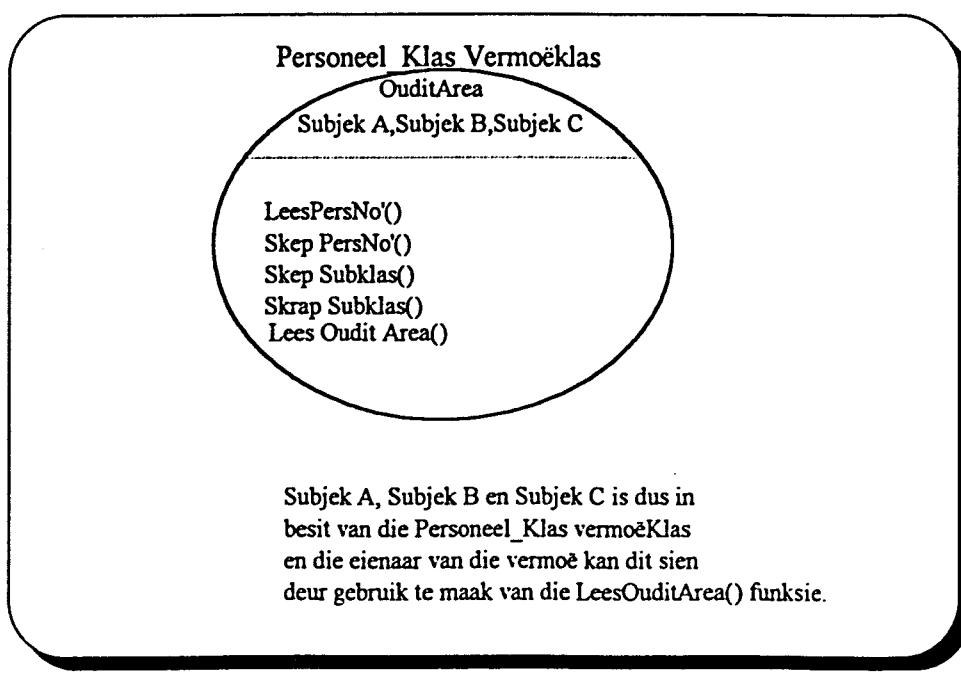


Fig 7.4. Die bywerking van die ouditarea van die ouerklas.

Daar moet egter ook beheermaatreëls toegepas word op die uitdeling van vermoëns. Dit word aanstons bespreek.

'n Ander belangrike faktor waarvan kennis geneem moet word, is die gebruik van "**SIGBAARHEID VAN BO**" (sien hoofstuk 5). Sigbaarheid van bo impliseer dat 'n subjek alle sub-entiteite of onderafdelings van sy entiteit sal kan sien en gebruik indien die subjek 'n vermoë besit vir die entiteit in geheel. Dit wil sê dat in die bogenoemde voorbeeld die personeelbestuurder, eienaar van die personeeldatabasis die vermoë wat aan hom gegee is as eienaar sal kan gebruik om toegang te verkry tot enige subklasse en objekte van sy personeelddatabasisklas

7.3. BEHEER OOR DIE UITDELING VAN VERMOëNS

In enige stelsel wat as veilig beskou word, is dit tog nie moontlik om die reg om toegangsreg te kan uitdeel, aan enigeen te gee nie. Die hoofklerk moet bv. nie die reg kan uitdeel aan 'n junior om enige salaris te lees nie, of om vermoëns tot die salarissubklas te kan uitdeel nie. In hierdie geval is daar beheer nodig oor die uitdeling van vermoëns.

Daar kan wel gebruik gemaak word van multivlak of verpligte sekerheidsbeginsels, maar dit word gelaat vir verdere studie in hierdie veld.

Voordat die uitdeelprobleem aangepak word, moet die konsep van die uitdeling van vermoëns eers duidelik wees en daarom sal daar nou eers aandag geskenk word aan die manier waarop vermoëns uitgedeel word, asook aan die manier waarop die reg om vermoëns uit te deel, aangegee word.

Die vermoëklas van die eienaar besit 'n funksie UITDEEL() wat hy kan gebruik om sy vermoëns uit te deel. Dié uitdeel()funksie kan sodanig verander word in 'n vermoësubklas dat dit die uitdeel van vermoëns slegs toelaat as dit

deur sekere subjekte gebruik word. Die eienaar kan byvoorbeeld in die uitdeelfunksie, rolle van subjekte spesifiseer waaraan vermoëns uitgedeel mag word of deur wie vermoëns uitgedeel mag word.

DIE UITDEEL VAN VERMOëNS

Beheer kan uitgeoefen word oor die uitdeel van vermoëns deur gebruik te maak van subjekrolle, of identifiseerders as beperkings in die uitdeelfunksie. Die gebruiker van hierdie beperkende maatreëls sal eerstens bespreek word en daarna sal aandag geskenk word aan die rol van die ouditarea in 'n vermoë.

Die eerste wyse waarop beheer uitgeoefen kan word oor die uitdeling van 'n vermoëns, is deur 'n vermoësubklas so te vorm dat vermoëns van hierdie subklas slegs aan sekere rolle of subjek identifiseerders uitgedeel mag word. Die Uitdeel()funksie in hierdie vermoësubklas sal die bogenoemde beperkings implementeer.

Die tweede wyse waarop beheer uigeoefen kan word oor die uitdeling van vermoëns en veral die gevalle waar 'n subjek eksplisiet geweier moet word om 'n entiteit te kan gebruik is deur, 'n eksklusief-uitsluitende vermoë te vorm en toe te ken aan die betrokke subjek.

'n Belangrike reël wat geld vir die bogenoemde is die reël wat spesifiseer of "Weiering van Magtiging"[Lunt] die voorreg bo "Magtiging" kry of die omgekeerde. DISMOD definieer hierdie reël soos volg :

Indien 'n subjek in besit is van 'n Eksklusief-uitsluitende Vermoë vir 'n spesifieke entiteit, sal alle toekenning van vermoëns (magtigings) tot daardie selfde entiteit oorheers word deur die eksklusief-uitsluitende vermoë. Dit wil sê, enige versoek vir die gebruik van 'n entiteit sal geweier word, indien die subjek in besit is van 'n eksklusief uitsluitende vermoë vir daardie entiteit. Die subjek sal weer die entiteit kan gebruik indien hierdie vermoë weggeneem is, maar die wegneemaksie kan slegs gedoen word deur die eienaar van 'n vermoë.

Die toekenning en wegneming van eksklusief-uitsluitende vermoëns kan slegs deur die eienaar van 'n entiteit gedoen word, tensy die eienaar van die entiteit spesifiek 'n vermoë uitdeel aan 'n ander subjek, wat die subjek in staat sal stel om hierdie funksies uit te voer.

DIE OUDITAREA

Die Ouditarea is die bergingsarea wat daar in elke vermoë bestaan met die doel om die eienaar en sub-eienaars van die entiteit op hoogte te hou van subjekte wat toegang besit tot die entiteit in hul besit.

'n Integrale gedeelte van die uitdeelfunksie is die gedeelte van die funksie wat die ***ouditarea*** van vermoëklasse en -subklasse bywerk. Dié gedeelte van die uitdeelfunksie sal die ouditarea bywerk met die identifiseerders van die subjekte aan wie vermoëns uitgedeel was. Die ouditarea kan op geen ander manier verander word as deur die uitdeel()- en wegneem()funksies nie.

Die ouditarea van alle klasse en subklasse in die hiërargie van die vermoëns van die eienaar, kan deur die eienaar van die entiteit gesien word. D.w.s. die eienaar van 'n entiteit is te alle tye bewus van wie toegangsreg tot sy entiteit besit. Die eienaar kan dan sy reg as eienaar uitoefen om 'n vermoë of vermoësubklas weg te neem indien dit nodig is.

'n Tipiese Ouditarea sal soos volg lyk :

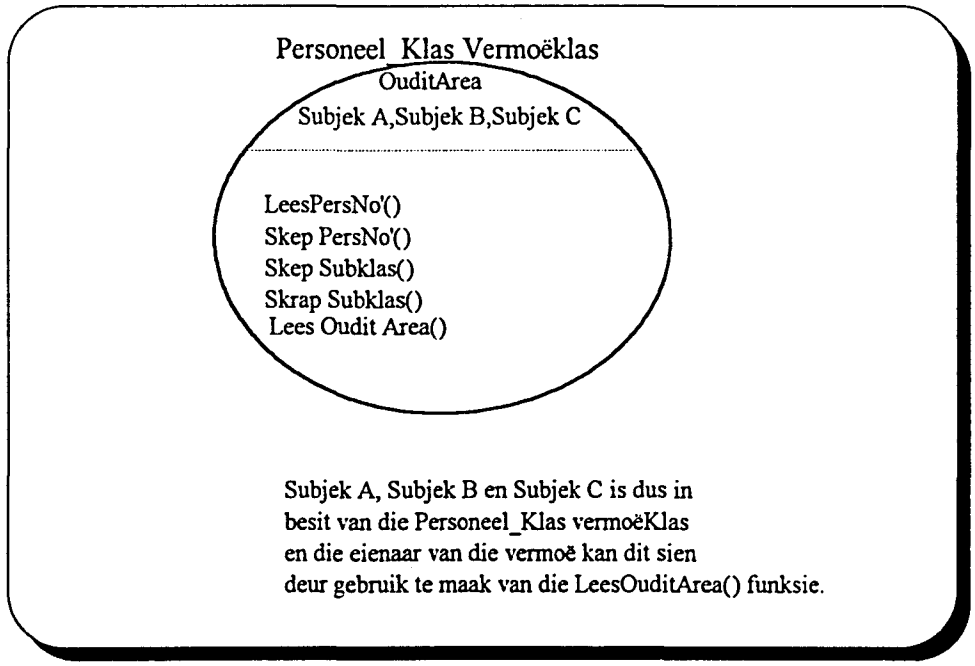


Fig 7. 5 Die Ouditarea.

Die UITDEEL VAN DIE UITDEELREG.

Sommige modelle maak gebruik van 'n aparte funksie vir die uitdeling van "UITDEEL-regte", d.w.s. die uitdeel van die reg om vermoëns te kan uitdeel, byvoorbeeld SEAVIEW wat gebruik maak van die funksies "GRANT" en "GIVE_GRANT"[Lunt]. DISMOD maak egter gebruik van die herdefiniëring van die uitdeel funksie in 'n vermoë om die funksionaliteit te kan verseker.

Die uitdeel van 'n uitdeelreg kan soos volg voorgestel word:

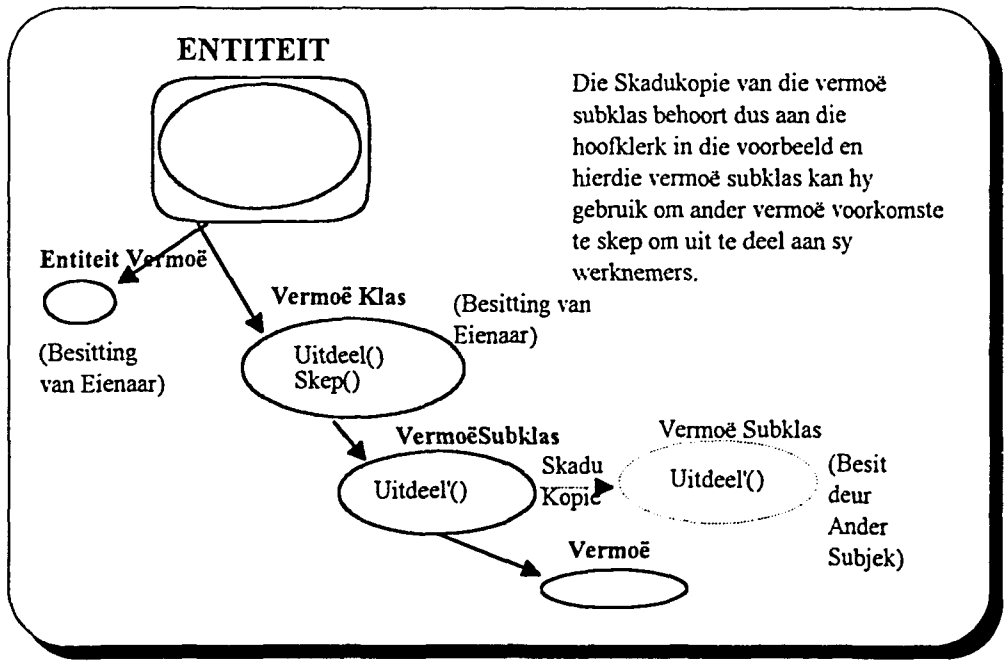


Fig 7.6. Die Uitdeling van 'n vermoësubklas.

Let op dat die uitdeel'()funksie in fig 7.6 'n geherdefinieerde uitdeel()funksie is wat die uitdeel() funksie as 'n "null" of "geen-waarde" funksie herdefinieer of spesifieke eienskappe koppel aan die uitdeel()funksie. Fig 7.6. toon dat met die uitdeel van 'n vermoësubklas, oftewel die "reg om vermoëns uit te deel", kan daar eerstens net 'n skadukopie van die vermoësubklas aan die sub-eienaar gegee. 'n ***Skadukopie*** is 'n wyser na die vermoësubklas van die eienaar se vermoë waarin die uitdeelfunksie bevat is. Die rede vir die gebruik van skadukopieë is dat dit makliker is om skadukopieë in stand te hou as werklike kopieë, want as die vermoësubklas verwyder word is alle skadukopieë onmiddellik ook vernietig, terwyl gewone kopieë dan eers opgespoor en vernietig moet word. Die gebruik van skadukopieë speel veral 'n belangrike rol in die wegneem van vermoëns.

Beheer word dus uitgeoefen met die uitdeel van vermoëns deurdat elke eienaar van 'n entiteit te alle tye weet wie toegang tot sy entiteit besit, en tweedens deurdat die eienaar van 'n entiteit in 'n vermoësubklas van die entiteit spesifiek kan spesifiseer aan watter subjekte betrokke vermoëns gegee mag word.

Die kontrole oor die wegneem van vermoëns word vervolgens bespreek word.

7.4. KONTROLE OOR DIE WEGNEEM VAN VERMOëNS

Dit is nou duidelik hoe vermoëns uitgedeel word maar dit is egter in baie gevalle nodig om vermoëns ook weg te neem.

Die wegneem van vermoëns kan geskied op drie wyses:

A As die gevolg van 'n funksie in die vermoë wat met 'n spesifieke tydsein of ander faktore geaktiveer word. In hierdie geval word die vermoë 'n ***self-vernietigende vermoë*** genoem.

B Met die stuur van die wegneemboodskap() aan die vermoë. Dié boodskap kan egter net deur die eienaar of sub-eienaar van die entiteit gestuur word.

C Met die stuur van die wegneemboodskap() aan die vermoë, maar in hierdie geval word die boodskap deur die stelselsekerheidsbeampte, of databasisadministrateur gestuur.

In geval A waar 'n vermoë wat self-vernietigend is, word 'n funksie in die vermoë ingebou wat die wegneemfunksie aan die vermoë self sal stuur sodat hy homself vernietig. Dit sal byvoorbeeld in die volgende geval gebeur:

Die Hoofklerk gee 'n vermoë aan een van sy klerke om tydens verhogingstyd die salaris_bywerk()funksie in die salarisklas te kan gebruik, maar nadat hierdie tydperk verstryk het, wil hy weer die funksie wegneem van die klerk af. Dit is egter nie nodig vir hom om te onthou om weer die vermoë te gaan wegneem nie, want hy kan die funksie in die vermoë inbou om homself te vernietig na die begrotingstydperk verstryk het.

Geval B, waar die eienaar die wegneemfunksie() aan 'n vermoë stuur. Dit sal byvoorbeeld gebeur wanneer 'n klerk sy diens beëindig. In hierdie

geval is dit nodig om die vermoëns wat aan hom behoort weer weg te neem sodat dit aan 'n ander subjek uitgedeel kan word. Dit mag ook nodig wees wanneer 'n subjek die entiteit verkeerdelik gebruik.

Dit is egter duidelik dat daar beheer moet wees oor wie die vermoëns mag wegneem en daarom bestaan die volgende reël :

Die wegneemfunksie mag slegs gebruik word deur die eienaar of sub-eienaar van 'n entiteit, sowel as deur die vermoë self.

Die moontlikheid van gebruik van entiteit sonder vermoëns sal vervolgens hanteer word.

7.5. DIE GEBRUIK VAN ENTITEITE SONDER VERMOëNS

Indien dit moontlik sou wees om enige entiteit te gebruik sonder die vermoë van daardie entiteit, dan sou dit nie nodig gewees het om 'n sekerheidstelsel te bou nie, daarom word maatreëls ingebou om hierdie moontlikheid te voorkom.

Die metode-identifiseerders in 'n entiteit is die kombinasie van 'n metode identifiseerder in 'n vermoë en 'n geënkripteerde sleutel in die vermoë. Beskou die volgende figuur as voorbeeld.

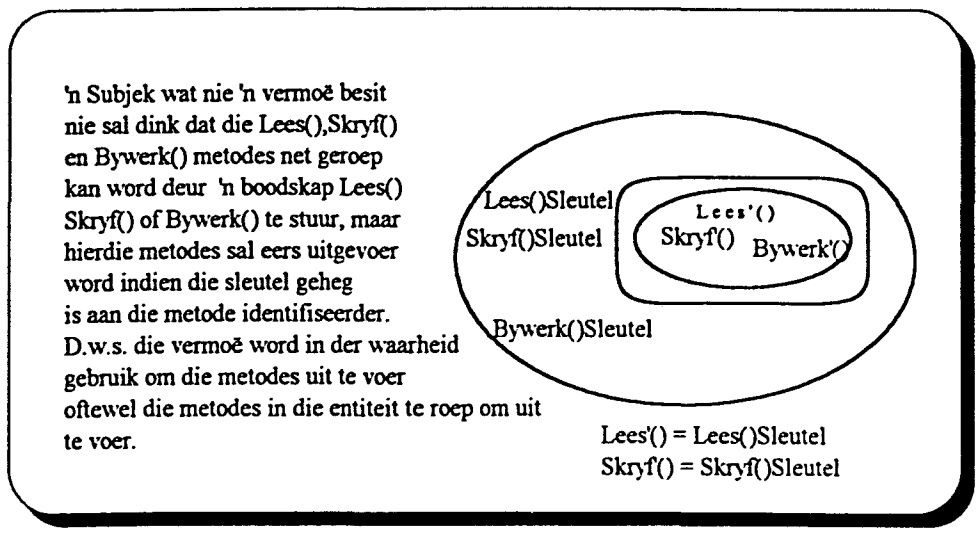


Fig 7.7. Die gebruik van 'n vermoë om metodes uit te voer.

Dit is duidelik uit die bogenoemde figuur (fig 7.7) dat as 'n boodskap gestuur word, dit eintlik deur 'n vermoë gestuur word. Dit werk soos volg:

A Die subjek wat die metode benodig vir uitvoering, stuur 'n boodskap aan die entiteit.

B Indien die subjek wat die boodskap stuur in besit is van 'n vermoë, word die vermoë bo-oor die entiteit gesit as sleutel en word die boodskap van die subjek verbind met die geënkripteerde sleutel van die vermoë. Die metode word nou geroep vanaf die vermoë met die korrekte identifiseerder van die metode.

C Die metode word nou uitgevoer as antwoord op die boodskap gestuur vanaf die vermoë en nie as antwoord op die boodskap vanaf die subjek nie.

Dit is dus onmoontlik om 'n boodskap vir die uitvoering van 'n metode te stuur indien die subjek nie in besit is van 'n vermoë nie.

7.6. BEHEERMOONTLIKHEDE VAN DIE EIENAAR VAN 'N ENTITEIT.

In die vorige gedeeltes is daar aandag geskenk aan die beheer wat uitgeoefen kan word met die uitdeel en wegneem van vermoëns. Ter afsluiting van hierdie gedeelte word daar 'n geheelbeeld gegee van die beheer wat 'n eienaar van 'n entiteit besit.

Watter beheer het die eienaar oor die gebruik van sy entiteite?

Uit die vorige afdelings kan ons aflei dat die eienaar beheer kan uitoefen oor sy entiteit op die volgende wyses :

A Die eienaar kan spesifiseer watter tipe toegang 'n subjek besit tot sy entiteit deur aan hom die toepaslike vermoë tot die entiteit te gee.

B Die eienaar kan beter beheer toepas deur die uitdeel- en skepfunksies in 'n subklas(geherdefinieer of nie) te gee aan 'n sub-eienaar van die entiteit wat in 'n mate beheer deel oor die entiteit.

C Die eienaar het die reg om te enigertyd 'n vermoë wat uitgedeel is vir hierdie entiteit weg te neem vanaf die subjek waaraan dit behoort, in die geval van misbruik of in die geval van sterfte of verdwyning.

D Die eienaar is ook ter enigertyd bewus van wie vermoëns tot sy entiteit besit omdat hy bewus is van wat in al die vermoëklasse en -subklasse van sy entiteit se ouditareas is.

Die eienaar is dus ten volle in beheer van sy entiteit maar kan ook beheer verdeel na sy sub-eienaars indien kontrolering van die entiteit te veel vir hom word.

AFSLUITING

Hoofstuk sewe het gepoog om die werking van DISMOD in meer besonderhede uit te lê. Hoofstuk sewe het veral gelet op die spesifieke beheermaatreëls wat toegepas kan word met die skepping van entiteite, soos byvoorbeeld die rol wat die stelselsekerheidsbeampte speel in die skepping van entiteite asook die beheermaatreëls wat uitgeoefen kan word in die skepping en gebruik van entiteite. Beheer oor die uitdeling en wegneem van vermoëns word ook breedvoerig bespreek. Probleme wat kan voorkom word ook in hierdie gedeelte behandel. Die rol wat die ouditarea in die vermoë speel word in die lig van uitdeling en wegneming van vermoëns bespreek. Ten einde laaste word die beheermoontlikhede van die eienaar van entiteite bespreek. Die verdere implikasie van DISMOD sal in Hoofstuk 8 bespreeks word.

---oOo---

HOOFSTUK 8

WERKING VAN DISMOD (vervolg)

8.1. INLEIDING

Hoofstuk 8 is 'n verdere uitbreiding op hoofstuk 7 en poog om die werking van DISMOD verder uit te lig. Daar sal veral gekyk word na die volgende onderwerpe, nl. :

Laat Binding

Eienaarvermoëns

Stelselsekerheidsbeampte se rol

Implikasies as subjek

Skadukopie (= Objek Identifiseerder)

Die gebruik van rolle.

Hoofstuk agt bespreek die werking van DISMOD verder en bied verskeie metodes waarop die stelsel geimplementeer kan word. Daar word veral gelet op die gebruik van verskeie objekgeoriënteerde programeringstegnieke wat gebruik kan word, soos byvoorbeeld die gebruik van laat binding, dinamiese binding, skadukopieë, ens. Die rol wat die sekerheidsbeampte kan speel word ook as opsionele eienskap van DISMOD bespreek.

8.2. Die STELSELSEKERHEIDSBEAMPTE

Wat is die werk van die sekerheidsbeampte of databasisadministrateur?

Die sekerheidsbeampte of databasisadministrateur is verantwoordelik vir die uitdeling van skeppingsregte sowel as die organisering en oudit van vermoëns. Die skeppingsregte is ingesluit in sy portefuelje, omdat daar iemand nodig is wat weet in watter area watter subjek werksaam is. Dit sal verhoed

dat enige subjek enige entiteit kan skep in enige area, waar die betrokke subjek byvoorbeeld nie die regte het om te werk nie.

Die bogenoemde aksies weerspieël slegs 'n gedeelte van die funksies van die stelselsekerheidsbeampte. Die verdeling van subjekte in gebruikersgroepe, asook die klassifisering van gebruikersrolle is nog 'n funksie wat sposifiek verrig word deur die stelselsekerheidsbeampte. Die bogenoemde taak kon deur beide die stelselsekerheidsbeampte of 'n databasisadministrateur verrig word, maar die laasgenoemde mag slegs deur 'n stelselsekerheidsbeampte uitgevoer word.

Die volgende gedeelte hanteer die reaksie van die sekerheidstelsel waar 'n gebruiker self kan aanteken of as deel van 'n groep kan aanteken.

8.3. DINAMIESE BINDING

Wat is Dinamiese Binding ?

Binding in die konteks van hierdie afdeling is die binding van 'n prosedureroep of -boodskap met die kode wat uitgevoer moet word as reaksie of respons op die prosedureroep of -boodskap. Dinamiese binding beteken dat die kode wat met die prosedure roep op boodskap geassosieer word slegs bekend word met uitvoertyd[CommACMv33]. Dinamiese binding word in die objekgeoriënteerde taal geassosieer met polimorfisme en oorerwing soos in die volgende voorbeeld duidelik gemaak sal word.

Indien daar 'n polimorfiese verwysing "personeel" bestaan met 'n statiese tipe van "Gebruikers", maar met 'n dinamiese tipe "Kontrakteur" en "Personeellid". Gestel dat daar 'n prosedure "Lees_Salaris()" is wat gedefinieer is in "Kontrakteur" maar wat geherdefinieer word in Personeellid as deel van die oorerwingafbeelding. Die Uitvoering van die Gebruikers.Lees_Salaris()roetine hang nou net af van die dinamiese tipe van "Gebruikers". As die dinamiese tipe "Kontrakteur" is, dan word die kode van "Lees_Salaris" in "Kontrakteur" gebind met boodskap, en as die dinamiese tipe

"Personeellid" is dan word die geherdefinieerde "Lees_Salaris()"funksie gebind met die boodskap.

Dinamiese binding speel 'n belangrike rol in DISMOD waar gebruikersrolle en identifiseerders 'n rol speel. Indien 'n gebruiker as deel van 'n gebruikersgroep aanteken, word die vermoëns van die groep met hom geassosieer en sal die metodes wat uitgevoer word deur die vermoëns van die groep gefilter word, d.w.s. die gebruiker sal die uitvoering van die boodskappe waarvoor hy vra deur die oë van die groep beskou.

Dit sal as volg geskied :

(A) 'n Gebruiker kan aanteken as die gebruiker self of as deel van 'n groep gebruikers.

Die eerste vraag wat gevra word, is :

As 'n gebruiker aan meer as een gebruikersgroep behoort, word die vermoëns van die gebruiker, die kombinasie van al die vermoëns van die gebruikersgroepe, of bly dit net die van een groep op 'n slag. Die keuse wat DISMOD as beleidsbesluit uitgevoer het is dat die gebruiker net die vermoëns van een gebruikersgroep op 'n slag kan gebruik.

(B) Die oomblik as die gebruiker aanteken, word die vermoëarea in die geheue vir die gebruiker opgebou, waarin alle vermoëns wat die gebruiker bevat, saamgevat word.

(C) Die vermoëns van 'n gebruiker bevat die dinamiese identifiseerder van die metodes waartoe die gebruiker magtiging besit met die vermoë as sleutel. Die vermoë dien ook as verdere filter indien daar ander funksies in die vermoë bestaan wat die betrokke entiteit sal beïnvloed soos gespesifiseer deur die skepper van die vermoë.

Die doel van die gebruik van dinamiese binding is om aan die eienaars van entiteite die vermoë te gee om aan enige ander subjek die spesifieke filter tot sy entiteit te gee, wat volgens sy diskresie korrek is. Dit word dus gedoen

deur slegs die vermoëns so te vorm dat die polimorfismes of dinamiese tipe identifiseerders van die entiteit in die vermoë ingebou word as sleutel of filter tot die entiteit.

Die implementeringsaspekte van vermoë behoort nou duidelik na vore te kom uit die bogenoemde bespreking. 'n Verdere aspek van die model wat die implementeringsaspekte duideliker behoort te maak is die gebruik van skadukopieë.

8.4. DIE GEBRUIK VAN SKADUKOPIEë

'n Vermoë uit die tradisionele oogpunt is die implementering van 'n ry in die toegangsmatriks wat soos volg lyk :

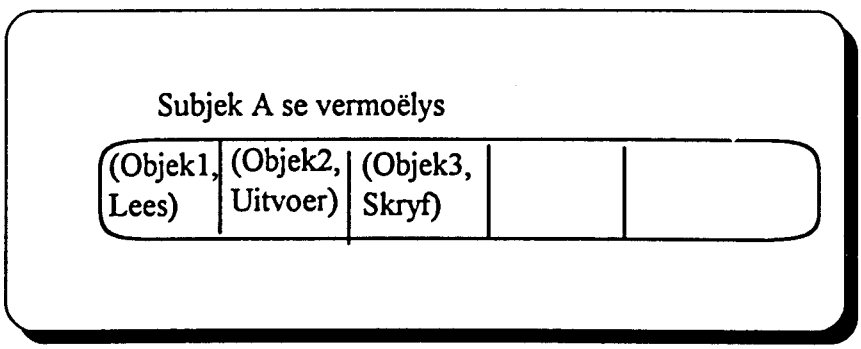


Fig 8.1. Die Tradisionele Vermoëlys

Die gebruik daarvan in DISMOD is dat daar in elke volgende vermoë-kompartement soos in fig 8.1 'n wyser na die vermoë objek sal wees wat die entiteit beskerm, of wat aan die betrokke subjek toegeken was. Dié wysers sal verwyder word met die wegneem van 'n vermoë.

'n Skadukopie is 'n wyser na 'n bestaande vermoë wat ook besit word deur ander subjekte. Die wyser bevat die identifiseerder van die toekenner-subjek en 'n wegneemfunksie kan net uitgevoer word deur die toekenner van die vermoë. Dit is maklik om hierdie vermoë weer weg te neem met die wegneemfunksie, omdat 'n lys van identifiseerders wat kopieë bevat in die

ouditarea gehou word en die identifiseerder kan dan net gebruik word om die wyser weg te neem uit die vermoëlys van die subjek wat die skadukopie bevat.

8.5. DIE GEBRUIK VAN ROLLE.

Rol-gebaseerde toekenning van magtiging mag die taak van die eienaar van entiteite baie vergemaklik. Afhangende van die vlak van 'n eienaar van 'n entiteit, mag die eienaar 'n rol-verdeling spesifiseer wat hy kan gebruik in die toekenning van die vermoëns tot sy entiteit. In die geval van die personeelbestuurder wat dit byvoorbeeld baie van pas, want hy het 'n entiteit op die hoogste vlak geskep, en baie ander objekte, subklasse en klasse kon in sy entiteit geskep geword het.

Indien die personeelbestuurder nou vooraf rolle gespesifiseer het soos administreerders, klerke, sekretaresses, ens. kon hy alreeds skepfunksies in sy vermoëklas gespesifiseer het wat slegs van toepassing sou gewees het op hierdie rolle. In die geval van gebruik van rolle moet daar aan elke subjek dan 'n rol gekoppel word, sodanig dat die uitdeler van vermoëns dan die vermoëns kon uitdeel aan 'n rol-groep eerder as aan subjekte. Alle subjekte wat dan die betrokke rol bevat, sal dan 'n skadukopie van die betrokke vermoë kry wat aan die rol-groep uitgedeel was.

Die gebruik van rolle is egter nog 'n baie vae area, daarom word hierdie gedeelte vir verdere studie gelaat.

8.6. GEVOLGTREKKING

Hoofstuk 8 het die gebruik van laat binding in DISMOD uitgelig en die rol van die stelselsekerheidsbeampte is in meer besonderhede verduidelik. Die gebruik van rolle om die uitdeel van regte te vergemaklik speel 'n belangrike rol in die nuwe sekerheidsmodelle wat ontwikkel word, en in DISMOD word daar ook uitgewys dat dit 'n belangrike rol kan speel in die vergemakliking van die uitdeel van regte aan subjekte.

Indeks

*

*-Eienskap, 24

A

Administrasie, 59
Aktiewe beskermingsmeganismes, 46

B

Bedreigings, 43
Beginsel van die maklikste indringing, 11
Beginsel van die minste voorreg,, 21
Beginsel van effektiwiteit, 11
Beginsel van tydloosheid, 11
Beleid van administrasie, 97
Beleidsrigting, 96
    Algemene Databasis Beleid, 97
Bell-en-Lapadulamodel, 23
Biba-model, 24

D

DAMOKLES, 88,120
Dataklassifikasie, 108
Databasisadministreerder, 109
Databasissekerheidsmeganismes, 59
    Gedesentraliseerde Beheer, 59
    Gesentraliseerde Beheer, 59
DISCO, 88, 93
Diskresionere sekerheid, 94,97,98,120,130
Diskresionêre sekerheidsmodel, 129
DISMOD, 129
Doel van die sekerheidsmodelle, 116
Domein, 46

E

Eenvoudige Sekerheidseienskap, 24
Eienaar, 47, 130,131,134
Eienaar-Subeienaarverhouding, 143
Eienaarskap, 59, 141
Eienaarskapbeleid, 97
EIUDM, 88
Eksklusief-uitsluitende Vermoë, 146
Eksplisiete Magtiging, 106
Eksplisiete Weiering, 106
Element van beskerming, 111
Entiteit, 45,131,132

F

Fabrikasie, 9
Formele sekerheidsmodel, 14
Formele spesifisering, 14

G

Gebruikersrol-gebaseerde sekerheid, 95
Geheuebeskerming 62
    -adresering, 62
Gidslys, 47
Graham-Denning Model, 25
Grein, 100,130

H

Harrison-Ruzzo-Ullman-model., 26
Honeywell Scomb, 34

I

Implisiete spesifiekheid, 107
Implisiete Weiering, 106
Inhoud-afhanklike toegangsbeheer, 100
Integriteit sekerheidsbeleid, 100
Interskakelende Erkende Siening, 41
ITSEC, 41

K

Klaringsvlak, 115
Klassifikasie, 22,115
Kompartemente, 22
Konfidensialiteit, 130
Konteks-afhanklike toegangsbeheer, 100
Kriptografie, 57

M

Maksimeer deling, 60
Meganismeseleksie, 14
Metodes, 133
Model vir magtiging, 113
Modelle,
    Enkelvlak, 16
    Inligting vloei, 17
    Multi-vlak, 16
    Reël-gebaseerd, 17
    Tralie, 16
MULTICS., 49
Multivlaksekerheid, 98, 120

N

Naam-afhanklike toegangsbeheer, 100
Negatiewe magtiging, 105
NETWERK BETROUBARE REKENAARBASIS (NBRB), 38
NETWERK TCSEC (TNI), 36
Nie-vervalsbare identitfiseerder, 134

O

Objekgeoriënteerde databasis, 129
Objekgeoriënteerde databasisse
    Gedrag-objekoriëntasie, 101
    Struktureel-objekgeoriënteerd, 95
Objekoriëntasie, 87
Objekte, 45,133
Oorerwing, 102
OSI - Die Oop Stelsel Interskakeling, 34
Ouditarea, 136,137,138,146

P

Passiewe beskermings meganismes, 46
PASSIEWE ELEMENTE, 111
Predikate, 107
Prosedure-georiënteerde Toegangsbeheer, 56

R

Reëls, 116
REGTE, 46
Rekenaarsekerheid-oorsprong, 8
Risiko-aanvaarding, 14
Risiko-evaluering, 14
Rol-gebaseerde sekerheid, 131

Indeks

Rol-gebaseerde toegangsbeheer, 95
Rolle van subjekte, 145

S

SEAVIEW, 126, 148
Sekerheid
    Definisie, 12
Sekerheidsbeleid
    Identiteit-gebaseerd, 16
    Reël-gebaseerde, 16
Sekerheidskern, 18
Sekerheidsplan, 14
Sekerheidsvereistes, 14
Self-vernietigende vermoë, 149
Sensitiwiteitsvlakreekse, 109
Sentrale VermoëLys, 55
SERTIFISERING, 65
    wagwoorde, 65
Sigbaarheid van bo, 102
Sigbaarheid van onder, 102
Skadukopie, 148
Skepvermoë, 136,141
Skepping van entiteite, 140
Sleutel tot die entiteit, 134
Slot- en Sleuteltoegangsbeheer Meganismes, 56
SODA, 88
Stelselsekerheidsbeampte, 130, 131, 138
Sub-eienaar, 152
Subjek, 47,131,133
Subklas, 152
supereienaar, 141

T

Take-Grant Stelsels, 27
TCSEC, 29
Toegangsbeheer
    Konteks-afhanklik, 61
    Naam-afhanklik, 60
Toegangsbeheerlys, 48
Toegansbeheermatriks, 50
    Eksklusief-uitsluitend, 51
TRUDATA, 126

U

Uitdeel()funksie, 145
UITDEEL-regte, 148
Uitdeling van vermoëns, 140
Uitleg van model, 110

V

Vermoë, 136, 141
Vermoë, 131, 134, 135
Vermoë, 51
    Berging, 52
    Probleme, 54
    Voordele, 54
    Werking, 53
Vermoë met Ouditarea, 137
vermoëklas, 145
VermoëLys, 55
Verpligte sekerheid, 123
Verwysingsmonitor, 19
Verpligte sekerheidsmeganismes, 130

W

Wegneem() funksie, 147
Wegneem van vermoëns, 140